

На DVD



Slackware 14

НА ВАШЕМ БЕСПЛАТНОМ DVD
Trisquel GNU/Linux
Плюс: AntiX, PDFы, коды к учебникам и более того!

LINUX FORM

Главное в мире Linux

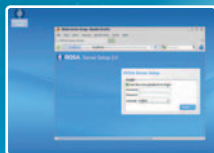
Январь 2013 № 1 (165/166)

Raspberry Pi

Сверхзадачи

- » Запись телепрограмм
- » Разгон процессора
- » Управление встроенными устройствами
- » Программирование огоньков для елки

Android И все-таки куб вертится
ZFS Внедряем в жизнь
Erlang Чем больше задач, тем лучше



ROSA Server

» Российский серверный дистрибутив на мировой арене **с. 17**



Преемник базы данных

« Руководители среднего звена — это именно они уничтожили Sun »

Скажите, что было не так, Монти Видениус **с. 44**

Кодирование Выбираем Geany

» Выбросьте старую IDE и ускорьтесь на 101%

Сайтостроение Резвый Django

» Web-приложение за несколько минут

Автоматизация дома OpenRemote рулит

» Управляйте бытовой техникой через web-интерфейс

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

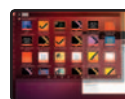
Linux center
www.linuxcenter.ru

ПЛЮС!
Пингвины
и дельфины
с. 42

Также в номере...

Капризы Кетсая

Ubuntu 12.10 готова. Поиски в Сети были ваши — станут наши **с. 14**



Bitcoin — чистая монета

Смоделируйте Великую экономическую депрессию 1929 года **с. 68**



Проект Byzantium

Защитим свою сеть от зловещих зомби **с. 56**

Патенты

» Клянусь бородой Зевса, что все это значит? **с. 48**



ПОДПИШИСЬ НА ЖУРНАЛ LINUX FORMAT!

2013 **LINUX**
FORMAT
Главное в мире Linux



Оформи в редакции
подписку на печатную
версию журнала
и получи в подарок
диск с архивом номеров,
а также подписку
на электронную версию
издания в формате PDF.

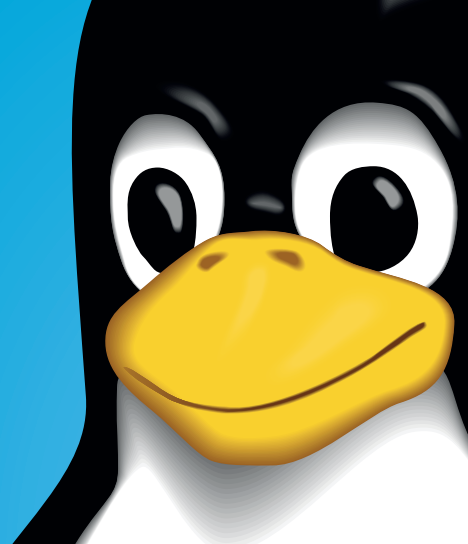


Стоимость подписки Годовая — 2280 руб., полугодовая — 1230 руб. без учета стоимости доставки.
Адреса и телефоны редакции Санкт-Петербург, Лиговский пр., 50, корп. 15, тел. (812) 309-06-86.
Москва, Красноказарменная ул., 17, тел. (499) 271-49-54.

Варианты доставки Почтой по России простой бандеролью — журнал доставляют прямо в почтовый ящик
» Почтой по России заказной бандеролью — в почтовый ящик приходит извещение, номера выдают на почте » Курьером
«ГНУ/Линуксцентра» по Москве и Санкт-Петербургу » Курьерской службой СПСР по России » В виде PDF-файлов
для подписчиков электронной версии.

shop.linuxformat.ru

Используйте свободное ПО — сэкономьте годовой бюджет!



Операционная система GNU/Linux поможет вам с **наименьшими затратами** решить проблему лицензирования программного обеспечения, навсегда избавиться от компьютерных вирусов и повысить надежность вашей компьютерной сети.



**С нашей
помощью
вы сможете**

**Сконцентрироваться
на своем бизнесе,**

не отвлекаясь на вопросы
поддержки своей
ИТ-инфраструктуры



**Забывать о вирусах,
угрозах безопасности**

и необходимости
лицензирования
программного обеспечения



**Оптимизировать
затраты**

на лицензирование ПО
за счет максимально
возможного использования
свободного ПО

ГНУ/Линуксцентр предлагает:

- внедрение наиболее дружелюбных вариантов ОС GNU/Linux и прикладных решений на базе свободного ПО;
- абонентскую поддержку вашей сети;
- обучение сотрудников вашей компании.

Наш опыт внедрения свободного программного обеспечения в организациях различного профиля поможет выбрать **оптимальное сочетание свободного и коммерческого программного обеспечения**, подходящее именно для вашей компании, а также поможет избежать технических и организационных проблем при внедрении свободного ПО.

**Решите проблемы лицензирования ПО и поддержки
компьютерной сети с помощью профессионалов!**

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

НАЙДИ РАБОТУ ЛЕГКО!

на www.hh.ru



САЙТ РАЗРЕШЕН ДЛЯ ПОСЕЩЕНИЯ БЕЗ ОГРАНИЧЕНИЯ ВОЗРАСТА

hh **ru**
HeadHunter

Выбирай из более чем 200 000 вакансий

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

Какие бы средства вы применили, чтобы оборониться от надвигающегося апокалипсиса широкомасштабных атак зомби?



Гэри Уокер

Я бы притворился одним из фанатов, которые торчат в очереди у магазина Apple. Цель зомби – мозги, верно?



Эндрю Грегори

У меня уже есть идеальное оружие, в виде моей увесистой клавиатуры из 1980-х.



Эфраин Эрнандес-Мендоса

Буду стоять на крыше своего дома, швыряя в них 18-ю дисками «Лучших баллад 1980-х», часть 6.



Бен Эверард

Я бы забаррикадировался непроданными журналами *Linux Format*.



Маянк Шарма

Установил бы гигантский экран ТВ-программы «Девчата» и пошел делать сэндвич. У зомби будет массовый суицид.



Джонатан Робертс

Обороняться?! Да я бы самолично проводил их к границе с Девонширом. Приятного аппетита, милашки...



Майк Сондерс

Наконец-то найдется применение куче красных ракушек, сваленных в моем гараже.



Валентин Синецын

За годы работы в Linux я немного соскучился по свежим 3D-стрелялкам. Где-то тут был мой дробовичок...



Ник Вейч

Примчу в Колумбию, перекину Кристину Агилеру через плечо и унесу ее в наше тайное любовное гнездышко под землей.



Сюзан Линтон

АК-47, кольт 1911, «Глок», браунинг, ракетница-переноска, термитные гранаты. Жду не дожидусь этого дня.



Шашанк Шарма

Выставил бы на улицу свой комп с окном настройки *MythTV* на экране. Это задержит их навсегда.



Нейл Боткин

Бутылек с дезодорантом и зажигалку.



ZX Spectrum XXI века

» В этом номере – сразу несколько статей о Raspberry Pi, проекте, который стремительно набирает популярность во всем мире. Секрет успеха прост – прекрасное соотношение цены и качества. Все близкие аналоги стоят как минимум в разы дороже.

Впервые удалось не только разработать, но и наладить крупносерийное производство недорогого компьютера (не дороже старших моделей Arduino), к которому легко подключить любые стандартные устройства ввода-вывода и на котором можно выполнять «обычную» ОС со стандартным же набором приложений. До сих пор выполнить все три условия одновременно не получалось: выходило либо дорого, либо нестандартно. Более того, подключить к нему различные датчики и схемы управления моторами почти так же легко, как к Arduino. Ожидается и появление плат расширения. Так что шансы превзойти по популярности легендарный ZX Spectrum вполне реальны.

Описание наиболее интересных применений Raspberry Pi можно найти на страницах нашего журнала и на сайте проекта, у которого есть и свой собственный журнал The MagPi (www.themagpi.com).

Создателям Raspberry Pi хочется пожелать в новом году дальнейших успехов. И чтобы единственной проблемой, как и в 2012-м, оставалось лишь большое число заказов.

А читателям LXF тоже пожелаем успехов, удачи и новых достижений в новом году!

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

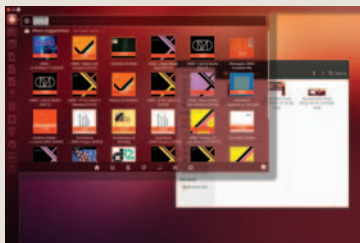
Содержание

Номер проходит под сладким знаком разлюли-малины.

Обзоры

Ubuntu 12.10 14

Почему Ubuntu следовало бы обойтись без версии номер 12.10.



➤ Кетсаль, может, и шустрый, но мог бы быть не таким корявым.

Slackware 14 16

Дедушка среди настольных дистрибутивов преспокойно гнет свою линию.



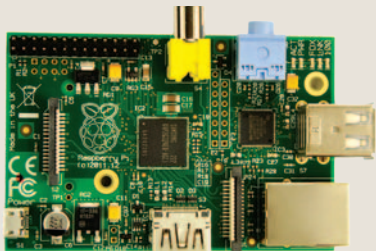
➤ Slackware 14 и Майк Сондерс: два надежных ветерана свободного ПО.

ROSA Server 17

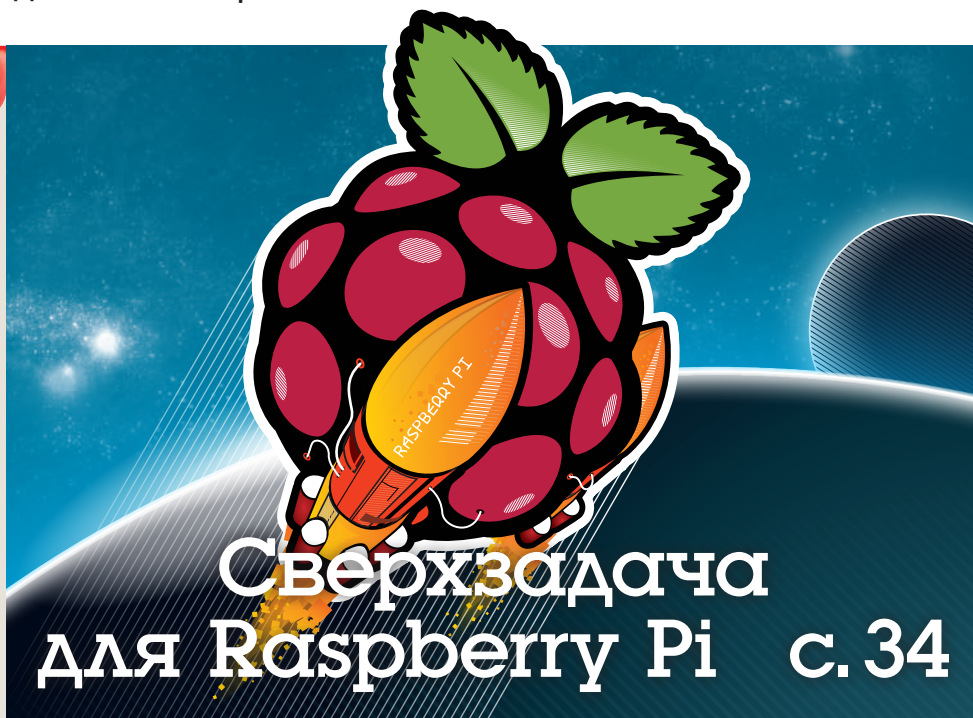
Из России с любовью... если ваш сервер есть в списке, настройка будет легка.

Raspberry Pi 512 MB 18

Потрясающий, инновационный, крошечный — все это уже было; но теперь еще и удвоена оперативная память!



➤ Raspberry Pi 512 MB: теперь с... ну вы поняли, с 512 МБ ОЗУ.



Сравнение: ПО для виртуализации с. 28



Что за штука — проект Byzantium? с. 56

Люди говорят



“ На верхнем уровне все было прекрасно. А ниже каждый лишь охранял свою кочку ”

Монти Видениус — про свою работу в Sun с. 44

На вашем бесплатном DVD



Slackware 14

» Старый соратник Linux

Trisquel

» Подлинно свободный дистрибутив Linux

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

Ищите в этом номере



Руководство по патентам **48**

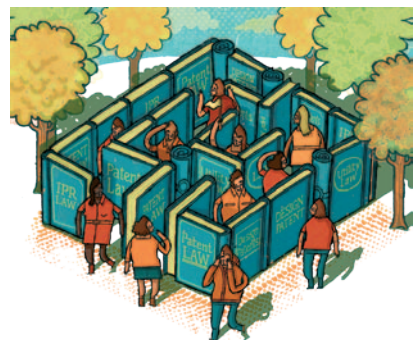
Иск Apple против Samsung – это просто.

ZFS в действии **52**

Применим перспективную файловую систему.

Сисадминам **58**

Не без блеска установим и защитим web-сервер.



Пропустили номер?

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



Учебники

Цифровая валюта Bitcoin **68**

Не храните деньги в банке, и в банках тоже не зарывайте: виртуальный компьютерный эквивалент гораздо надежнее.

Web-разработка Django **72**

Лихо прокатимся на Ferrari фреймворков разработки web-приложений.

Программирование Geany **76**

Разберемся, почему Geany является идеальным редактором для программистов.

Администрирование Udev **80**

Чудо из чудес: одновременно и динамическая файловая система, и менеджер устройств.

Безопасность SSH **82**

Протокол Безопасной оболочки как броня для ваших сетевых соединений.

Автоматизация OpenRemote **84**

Пусть к вашему возвращению домой телевизор сам включится, а пироги скакнут в печку. Без чужьего велья!

Raspberry Pi Строим дистрибутив **88**

Держите свой Pi под полным контролем, создав собственный дистрибутив Linux.

Языки программирования Erlang **92**

Многозадачность настоятельно требует упражнений – одного практикума явно мало.

Постоянные рубрики

Новости **4**

РОСА подружилась с автопромом, новая мультка – ультрабук, успех СПО в Европе переменный, Столлмен недоуволен Ubuntu, а Linux получит игровую консоль.

Новости Android **20**

MeeGo бросает вызов, совместимость взята под защиту, киберсквоттера пресекли, а Yota выпускает смартфон с двумя экранами.

Android **22**

Джульетте Кемп показалось обидным, что на Android нет вращающегося куба, и она принимается осваивать OpenGL.

Сравнение **28**

Лучшие приложения для виртуализации на настольном компьютере.

Что за штука **56**

Наладьте связь с другими целевыми после массовой атаки зомби при помощи проекта Byzantium.

Рубрика сисадмина **58**

Наш путь достиг вершины. На прощание углубимся в ядро и в проблемы безопасности.

Ответы **96**

ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!

Нейл Ботвик отвечает на вопросы про Android, KMail, разрешение экрана и разбиение диска.

Hotpicks **100**

Отведите горяченького: лучшие в мире новинки свободного ПО.

Диск Linux Format **106**

Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... **108**

Если бы можно было повернуть время вспять... Но не все потеряно!

Через месяц **112**

Узнайте, почему Linux Mint еще покруче вашего будет.



ГЛАВНОЕ Немецкий опыт » Роса в машине » Спутник программера » RMS vs Canonical
» Консоль от Valve

В ЕВРОПЕ ВНЕДРЯЮТ СПО

Переменный успех перехода

Где-то получается сэкономить деньги, где-то происходит разочарование и откат к «проприетарщине».



» Рубрику готовил
АРТЕМ ЗОРИН

Европейцы часто идут на эксперименты, иногда удачные, а иногда сомнительные. Многие европейские страны и города уже не первый год пытаются внедрить свободное ПО в систему государственного управления. Кое-где это даже получается – например, в немецком городе Мюнхене. Исследование финансовой эффективности перехода госучреждений Мюнхена на Linux и СПО показало, что данная инициатива позволила городу сэкономить от 8 до 12,8 млн евро бюджетных средств. Результаты сравнительного расчета бюджета были оглашены независимой группой Free Voters на состоявшемся недавно заседании городского комитета по информационным технологиям.

В основу расчета было положено сравнение совокупного расходования бюджетных средств при использовании Windows и миграции на дистрибутив LiMux, представляющий собой специально адаптированную для использования в муниципальных учреждениях редакцию Ubuntu Linux. Было рассмотрено два технологически эквивалентных сценария использования Windows: Windows + Microsoft Office и Windows + OpenOffice, которые обошлись бы казне на 12,8 и 8 млн евро дороже, чем LiMux.

При проведении исследования учитывался опыт миграции на Linux около 11 тыс. рабочих мест в администрации города и задействования свободного офисного пакета приблизительно на 15 тыс. рабочих мест. В итоге были получены следующие цифры: для конфигурации Windows с Microsoft Office связанные с операционной системой расходы составили €11,6 млн, для конфигурации Windows и OpenOffice – €7,4 млн. Из них покупка и подписка на обновления Microsoft Office обошлась в €4,2 млн, а Windows – в €2,6 млн. Дополнительно около €5 млн

пришлось бы потратить на покупку нового оборудования, для замены устаревших систем, не соответствующих требованиям Windows 7. Стоимость миграции приложений оценена в €55000.

В ситуации использования LiMux отсутствовали затраты на приобретение лицензий на ПО, а также необходимость замены устаревших рабочих станций, которые укладывались в минимальные требования к оборудованию, предъявляемые дистрибутивом LiMux. К сентябрю 2012 года дополнительные расходы проекта на основе LiMux составили только €270 тыс. Данные расходы были связаны, как правило, с миграцией приложений.

Суммарная стоимость решения на базе Windows и Microsoft Office составила €34 млн, Windows и OpenOffice – €30 млн, LiMux – €23 млн.

Это показатель того, что при грамотном подходе можно мигрировать на Linux, стоит только этого захотеть и не бояться.

Однако не все и не всегда так хорошо, как кажется на первый взгляд.

К концу 2012 года стало известно, что Администрация немецкого города Фрайбург приняла решение вернуться к MS Office после попытки внедрения в документооборот OpenOffice, оказавшейся неудачной: она вызвала большое количество жалоб со стороны пользователей и серьезные проблемы с совместимостью



» Это Администрация города Фрайбурга. Здесь отказались от OpenOffice.

с проприетарными решениями. Для утверждения решения было проведено голосование – в результате 25 членов совета проголосовали за полный отказ от OpenOffice, 2 воздержалось, 20 были против.

Немцы рассматривали возможность перехода на OpenOffice с 2007 года, но возложенные на него надежды себя не оправдали. Было решено, что совместное использование OpenOffice 3.2.1 в комбинации с Microsoft Office 2000 приведет к ухудшению производительности работы, нарушению связей с внешними организа-

«Винить в “криворукости” пользователей совсем неправильно.»

циями и недовольству пользователей. Сотрудники администрации жаловались на проблемы с совместимостью со всеми форматами Microsoft Office – Word, Excel и PowerPoint. Взвесив все варианты, администрация пришла к решению о повсеместном переходе на Microsoft Office 2010.

Не всегда СПО может быть внедрено и применяться с успехом: достаточно вспомнить «школьный проект» нашего государства, когда насильно пытались «внедрить» Linux. Проект умер, так толком и не родившись. Каждому инструменту – свое применение, и в том, что люди разочаровываются в СПО, нет ничего страшного: СПО само по себе является не целью, а лишь средством достижения цели – экономии денег, и это самое главное. Видимо, у работников Администрации Фрайбурга не получилось сэкономить, внедрив СПО. И винить в «криворукости» пользователей совсем неправильно – виной всему, скорее всего, недоработанность самих свободных программ.

XV НАЦИОНАЛЬНЫЙ ФОРУМ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

ИНФОФОРУМ 2013

ЗДАНИЕ ПРАВИТЕЛЬСТВА МОСКВЫ,
НОВЫЙ АРБАТ, 36

5–6 февраля

КОНФЕРЕНЦИИ

ЭКСПОЗИЦИЯ

ПРЕМИИ

МАСТЕР-КЛАССЫ

Крупнейшее
событие отрасли
информационной
безопасности



КОНФЕРЕНЦИЯ «БЕЗОПАСНЫЙ ИНТЕРНЕТ»

- Киберпреступность в Интернете
- Безопасность социальных взаимодействий через Интернет
- Мастер-классы

КОНФЕРЕНЦИЯ «БЕЗОПАСНЫЙ ГОРОД»

- Безопасный город: формирование концепции (заседание Экспертного совета Комитета Госдумы по безопасности)
- Комплексные проблемы и инновационные решения

КОНФЕРЕНЦИЯ «ПЛАТЕЖНАЯ СИСТЕМА И ЭЛЕКТРОННАЯ ТОРГОВЛЯ»

- Национальная платежная система России: проблемы ИБ
- Информационная безопасность в технологиях ДБО

ДИСКУССИОННАЯ ПРОГРАММА

- Защита персональных данных
- Практика реализации электронных услуг населению
- ЦОДы, «облака» и безопасность
- Кадровый заказ в сфере ИТ и ИБ

*Ведущие специалисты отрасли, участники из регионов России и стран СНГ;
площадка для эффективного диалога госзаказчиков и разработчиков;
проверенные и перспективные информационные решения;
интересные партнерские программы и выгодные условия участия.*

**НОВЫЙ ГОД ИТ-МЕРОПРИЯТИЙ ТРАДИЦИОННО
НАЧИНАЕТСЯ С ИНФОФОРУМА. БУДЬТЕ С ПЕРВЫМИ!**

Инфофорум проводится при поддержке Комитета Государственной Думы РФ по безопасности и противодействию коррупции, Аппарата Совета Безопасности РФ, Минкомсвязи России, МВД России, Минобороны России, ФСБ России, ФСТЭК России, Роскомнадзора и других министерств и ведомств.

Программа и регистрация
участников на
www.2013.INFOFORUM.RU

LINUX В ВАШЕМ АВТО

ИТ и автопром

РОСА вошла в группу разработки информационно-развлекательных систем.

Компания ROSA, разрабатывающая серию отечественных дистрибутивов Linux, объявила о вхождении в руководящий комитет рабочей группы Automotive Grade Linux, созданной организацией Linux Foundation с целью разработки эталонной Linux-платформы для автомобильных информационно-развлекательных систем, интерактивных приборных панелей, встроенных информационных систем и других автомобильных устройств.

В состав рабочей группы входят ведущие мировые производители автомобилей, лидеры рынка потребительской электроники и крупные ИТ-компании: Jaguar, Land Rover, Nissan, Toyota, HARMAN, Intel, Samsung, Fujitsu, NEC, NVIDIA и TI.

В данный момент время компания ROSA ведет разработку информационно-развлекательной системы ROSA Sputnik, сочетающей функции автомобильного навигатора и мультимедиа-центра. Кроме того, компания проводит исследования

в области построения семантических интерфейсов и оптимизации организации и доступа к информации.

Помимо классических критериев (мощность, скорость, экономичность, безопасность), покупатели все чаще оценивают автомобили с точки зрения их информационно-развлекательных возможностей. «Избалованные» прогрессом коммуникационных устройств потребители ожидают получить ставшие привычными технологии и возможности в своем новом автомобиле.

«ROSA» смогла добиться хороших результатов в разработке программного обеспечения для компьютеров и мобильных устройств. У нас есть видение того, как сделать эти наработки полезными и привлекательными для автомобилистов», утверждает директор по развитию международного бизнеса и партнерским программам «РОСЫ» Владимир Крюков. В частности, на выставке в Лондоне осенью 2012 года «РОСА» показала прототип

Интерфейс автомобильной информационно-развлекательной системы ROSA Sputnik.



информационно-развлекательной системы ROSA Sputnik. Действующий макет позволяет проигрывать мультимедиа, знакомиться с достопримечательностями с помощью экскурсионных модулей и прокладывать маршруты с учетом большого количества критериев.

Вся информация может быть выведена как на экран бортового компьютера, так и на лобовое стекло автомобиля. Для этого были разработаны модули интерфейса с использованием принципов Augmented Reality.

Приятно, что российские программисты получают международное признание.

ЕЩЕ ОДИН ИНСТРУМЕНТ РАЗРАБОТЧИКА

Новая игрушка для умников

Компания Dell выпустила для разработчиков ультрабук на базе Ubuntu.

Dell вывела на рынок первый ультрабук, разработанный в рамках проекта Sputnik – его целью является создание специального профессионального ноутбука для разработчиков сетевых приложений и ПО для мобильных и облачных систем. Устройство базируется на уже выпускаемом ультрабуке Dell XPS13 с 13,3-дюймовым экраном (1366×768) и весом всего 1,36 кг. Размер ультрабука – 31,6×20,5 см, толщина – 6–18 мм. Процессор – Intel Core i7-3517U, видеокарта Intel HD 4000, 8 ГБ ОЗУ и 256-ГБ SSD-диск. Ультрабук стоит \$1449 – на \$50 меньше, чем аналог с Windows 8, а более простые модификации продаются за \$999.

Одновременно вышли в продажу недорогие ноутбуки Vostro 2520 (экран 15,3 дюйма, CPU Celeron Dual Core, 2 ГБ ОЗУ, 320 ГБ НЖМД); в поставке с Ubuntu продаются по цене \$299, а аналогичная модель с Windows на \$70 дороже.

ПО ультрабука Sputnik основано на свободной для загрузки сборке Ubuntu 12.04, адаптированной для поддержки аппаратных компонентов устройства и расширенной средствами формирования различных классов окружений для разработки приложений. Для распространения модифицированного пакета с ядром Linux введен PPA-репозиторий. Сборка для Sputnik (1,3 ГБ) почти вдвое больше обычного установочного диска Ubuntu.

Особенность этой системы – поставка утилит для работы со специально созданным в GitHub репозиторием, где размещен набор профилей для различных платформ разработки, позволяющих быстро сформировать среду для разработки.

Кроме того, в систему добавлены средства для локального тестирования и развертывания решений для облачных систем, основанные на идее запуска локального «микрооблака» на базе об-



Ультрабук Sputnik – красивое дитя сотрудничества Canonical и Dell.

лачной инфраструктуры OpenStack. После тестирования продукта в «микрооблаке» возможно его автоматизированное перемещение во внешние облачные системы, такие как EC2 от Amazon. Данные средства базируются на развиваемом компанией Canonical инструментарии Juju.

Dell сейчас занимает вторую позицию на рынке продаж ПК, уступая лишь HP.

softline®



Services

Software

Cloud

ИТ-архитектура вашего бизнеса



КАКАЯ СВОБОДА ЛУЧШЕ НЕСВОБОДЫ?

Патриарх СПО критикует дистрибутив № 2

Нарушение приватности – вот главная претензия Ричарда Столлмена к разработчикам.

Ричард Столлмен написал статью с резкой критикой Ubuntu, в которой сравнил применяемые в этом дистрибутиве методы со шпионским ПО, а также порекомендовал энтузиастам свободного ПО прекратить распространять Ubuntu и даже не советовать данный дистрибутив пользователям. Столлмен говорит, что значительное преимущество СПО в том, что сообщество защищает пользователей от вредоносного ПО; Ubuntu же,

запросов в панели Dash. Главное недовольство было связано с тем, что данная функция включалась по умолчанию, причем пользователь явно не предупреждался, что его данные без предварительного согласия отправляются сторонней компании. По мнению Столлмена, пользователь должен иметь возможность отказаться от отправки данных на внешние службы для любого запроса, чего можно добиться, добавив отдельную кнопку, по которой всегда будет осуществляться только локальный поиск информации.

Столлмен недоволен также тем, что в поставку дистрибутива включаются несвободные программы или предлагается их установка, а от имени Ubuntu рекомендуются и рекламируются несвободные продукты. Из негативных моментов также упоминается ограничивающая политика Canonical в отношении торговой марки Ubuntu, запрещающая коммерческое распространение копий Ubuntu под тем же именем, что является существенным урезанием свободы пользователей.



► Ричард Столлмен (слева) не видит или не хочет видеть реалий современности, а Джоно Бэкон (справа) считает его упреки однобокими.

Такие явно коммунистические высказывания Столлмена не остались без реакции со стороны Canonical. В блоге компании появилось сообщение менеджера Canonical по работе с сообществом Джоно Бэкона [Jonno Bacon], в котором тот назвал сообщение Столлмена необоснованным, однобоким и излишне категоричным, не считающимся с точкой зрения другой стороны и нагнетающим негатив.

По мнению Бэкона, понятия свободы и приватности сильно зависят от индивидуального мировоззрения отдельного человека. У разных людей разное отношение к приватности: одни считают, что должны полностью контролировать все аспекты своего присутствия в Сети, другим не важно, что информация об их посещениях и активности остается в чьих-то руках.

Разработчики Ubuntu пойдут дальше по пути интеграции дистрибутива с внешними службами, выйдя далеко за рамки добавленной в Ubuntu 12.10 функции показа предложений о покупке фильмов в Amazon. Общая идея, которая продолжает свою реализацию в Ubuntu 13.04, связана с использованием Dash как основного средства поиска любой информации, интересной пользователю, и стиранием грани между тем, где именно хранятся данные пользователя. Например, при попытке поиска музыки, книги или фильма предложение купить искомым материал представляется логичным. Для тех, кто не желает использовать функции обращения к внешним службам, в настройках предусмотрена возможность ограничить область поиска локальным компьютером.

«Причиной резкой критики стали проблемы с приватностью.»

по его мнению, сейчас можно рассматривать как пример обратных действий.

Причиной резкой критики Столлмена стали проблемы с приватностью в Ubuntu 12.10, на которые ранее уже обращала внимание организация Electronic Frontier Foundation. Напомним, что в Ubuntu 12.10 была добавлена функция показа контекстной рекламы, осуществляющая отправку в сервис Amazon приватных данных, фигурирующих в процессе ввода поисковых

Зачем такая свобода?

В Canonical утверждают, что компания прилагает все усилия для сохранения приватности, считает данную область приоритетной и уже на протяжении многих лет демонстрирует на практике свою лояльность пользователям при поддержке служб и сайтов.

В случае с онлайн-поиском в Dash все фигурирующие в поисковых запросах данные передаются через специальный транзитный сервер Canonical, который позволяет обезличить данные и исключить возможность сопоставления запроса с конкретным пользователем или отслеживания предпочтений отдельных пользователей.

Canonical, безусловно, развивается в правильном направлении, предоставляя все больше возможностей интеграции настольной системы с Сетью и монетизируя Ubuntu, а высказывания Ричарда

Столлмена, при всем уважении, не выдерживают конструктивной критики. Самое смешное, что и у таких некорректных высказываний в сообществе найдутся сторонники – истинные фанатики, готовые в ущерб свободе выбора выбирать весьма сомнительную «свободу» не иметь возможности покупать легальный контент прямо с рабочего места... Судя по тому, куда движется разработка Ubuntu, дистрибутив скоро станет уникальной площадкой для тестирования не только нового ПО, но и новых методов заработка на СПО, которых доселе очень мало: продавать будут контент и рекламу.

Несмотря на все вышесказанное, Ричард Столлмен был, есть и будет тем, кто дал толчок развитию всему сообществу СПО; однако и маститый гуру не всегда и не во всем бывает прав.

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

ИГРЫ К НАМ ПРИХОДЯТ

Steam для Linux и не только

Valve планирует выпустить собственную игровую консоль на базе Linux.

Гэйб Ньювелл [Gabe Newell], управляющий директор компании Valve, подтвердил информацию о планах Valve по производству собственной игровой консоли на базе Linux, которая будет конкурировать с консолями нового поколения от компаний Sony и Microsoft. Ньювелл отметил, что консоль будет лишена определенной гибкости и ограничена достаточно узким кругом решаемых задач, так как устройство не позиционируется в качестве замены полноценному настольному компьютеру.

В частности, консоль будет сочетать в себе функции медиа-центра и будет ориентирована на установку в качестве телеприставки в гостиных.

По словам Ньювелла, в следующем году ряд компаний планирует выпустить на рынок специализированные ПК для гостиных, на которых в том числе может быть предустановлен клиент *Steam*. Со своей стороны, компания Valve намерена создать собственное целостное программно-аппаратное решение.

Те, кому функции консоли Valve покажутся недостаточными, смогут остановить свой выбор на ПК от другого производителя, которые могут быть основаны на стандартном графическом стеке Linux и ПО для доступа к *Steam*. Подобным шагом Valve планирует сформировать новую игровую экосистему, не зависящую от отдельных производителей оборудования и программного обеспечения.

В то же время, Valve объявила о расширении программы бета-тестирования Linux-версии клиента сервиса доставки игр *Steam*. В частности, в декабре отправлена новая большая порция приглашений для желающих принять участие в тестировании (точная цифра не сообщается, но в ноябре/декабре 2012 г. было отправлено 5000 приглашений).

Также заявлено о создании новых форумов для пользователей Linux (например, для обсуждения работы игр на систе-

«Число игр, помеченных как поддерживающие Linux, возросло.»

мах с графическими картами AMD/NVidia/Intel), расширении возможностей системы информирования о проблемах. Число игр, помеченных в каталоге *Steam* как поддерживающие Linux, возросло с 26 до 32.

Аппаратные требования для запуска игр в Linux в целом не отличаются от аналогичных требований для платформ Windows и Mac OS X.

На наших глазах происходит превращение Linux в игровую платформу. Быть может, это и станет расцветом Linux «на десктопе», о котором так много и долго говорили все участники сообщества? Скорее всего, да. Именно так и должно быть, если верить новостям из Valve. **LXF**



➤ Гэйб Ньювелл (второй слева) как бы намекает нам, что все у нас будет хорошо...

Новости короткой строкой

➤ Согласно исследованию университета г. Санта-Клара (Калифорния), все патентные разбирательства в США инициированы патентными троллями. Источник: <http://www.opennet.ru>

➤ Линус Торвалдс осуществил слияние текущей активной ветки разработки ядра Linux с веткой, в рамках которой ядро избавлено от поддержки процессоров 386-DX/SX. Источник: <http://git.kernel.org>

➤ Большинство голосов городской совет Берна (Швейцария) утвердил новую IT-стратегию муниципалитета, нацеленную на переход госструктур города на свободное и открытое ПО. Источник: <https://joinup.ec.europa.eu>

➤ Компания PengPod сообщила об успешном завершении кампании по сбору средств для начала производства планшетного ПК с одно-временной предустановкой двух систем: Android 4 и KDE-оболочки Plasma Active. Устройство вот-вот поступит в продажу. Источник: <http://pengpod.com>

➤ Разработчики компьютера Raspberry Pi анонсировали скорое начало массового производства модели A, которая будет поставляться по цене \$25, но будет близка по своим характеристикам к первой редакции модели B, продаваемой за \$35. Источник: <http://www.raspberrypi.org>

➤ Некоммерческая организация KDE e.V., занимающаяся технической, юридической и финансовой поддержкой проекта KDE, отметила свое пятидесятилетие. Источник: <http://ev.kde.org>

➤ 11 декабря 2012 г. Intel представила семейство процессоров Intel® Atom™ S1200 – первые в мире маломощные 64-разрядные серверные однокристальные системы для микросерверов, с высокой плотностью размещения вычислительных ресурсов. Это новый класс энергоэффективных сетевых систем и систем хранения данных. Источник: intel-news@marketingcentre.ru

➤ Учебный Центр Инвента совместно с Представительством компании Red Hat в России провел 15 декабря 2012 года второй бесплатный семинар из серии «Red Hat: продукты и технологии». Серия семинаров предназначена для технических специалистов – администраторов Linux-систем. Источник: <http://rhd.ru>



АЛЕКСЕЙ ФЕДОРЧУК
Тэг <сарказм>
по умолчанию,
смайлики по вкусу.

О причинах systemd'изации

Феномен тотального внедрения менеджера инициализации *systemd* требует своего объяснения. И самое простое из них – методом аналогии.

Вспомним стенания о несовершенстве законодательства, раздававшиеся со всех концов нашей страны на заре ее капитализации. Для исправления чего за последние 20 лет законов было принято больше, чем за все годы Советской власти. Законов на все случаи жизни – от регулирования Интернета до отношения к «животным-компаньонам». Что, впрочем, не сделало их исполнение более обязательным. Потому что никому из наших законодателей не пришло в голову выступить с инициативой: наложить мораторий на принятие новых законов лет на десять. И попробовать исполнять хотя бы часть законов, уже существующих.

Аналогичная ситуация нынче сложилась и вокруг Linux'a. «Законотворческих» инициатив к исходу нулевых было накоплено... целые геологические напластования. Осталось только применить их в разработке готовых решений. Однако начался новый виток инициатив. И в результате время решений опять откладывается. Вероятно, до той поры, пока современные инициативы не покажутся устаревшими, и их надо будет заменять более прогрессивными.

А заниматься готовыми решениями – это ведь ничуть не менее скучно, чем добиваться исполнения уже существующих законов вместо принятия новых.

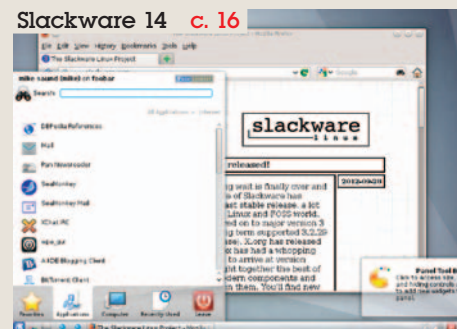
alv@posix.ru

Сегодня мы рассматриваем:

- Ubuntu 12.10** 14
Народный дистрибутив, самый лучший способ приобщения к Linux для новичков, держал победу в кулаке. А затем... затем он ввел кучу недопеченных функций, нарушил совместимость со старым оборудованием и кидал идеи об стену, пока не прилипли самые отстойные. Позор джунглям.
- Slackware 14** 16
Старый конь борозды не портит. Нашему алфавиту и цифрам не одно столетие, и тем не менее их применения продолжают блистать новинками. То же и с технологиями: если они работают, то неважно, сколько им лет. Если вы любите добротные вещи и не возводите погоню за новизной в культ, попробуйте Slackware. Это антик!
- Rosa Server 2012** 17
Прелесть свободного ПО в том, что проекту никогда не грозит кончина, если достаточное количество талантливых и трудолюбивых людей хотят его сохранить. ROSA – тот самый случай: это продолжение блистательного дистрибутива Mandriva, переориентированного на серверы группой русских программистов. Эпохальная победа.
- Raspberry Pi 512MB** 18
Манна небесная! Питай нас, доколе мы не насытимся! А затем добавь малинового варенья и малость ОЗУ, ибо мы взбалкали. Правду сказать, мы немало порадовались, что у кого-то хватило разума построить шоссе M4, по которому ныне доставляют новые уэльские Pi. Спешите купить!



Мы бы покупали у Amazon больше, если бы его трехмиллионные продажи давали бы отчисления на развитие СПО.



[Подобрать голос диктора] Человек... один в целом мире... с tar-архивами и текстовым редактором...

Сравнение: Приложения для виртуализации с. 28

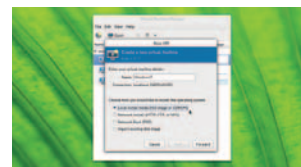
Parallels Workstation



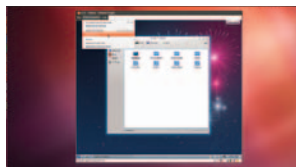
VirtualBox



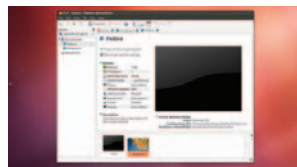
VM Manager



VMware Player



VMware Workstation



Магия виртуализации позволяет обрести силы на эксперименты с системой без боязни ошибок, которые обходятся столь дорого. И какой же инструмент виртуализации более всего подходит лично вам?

Ubuntu 12.10

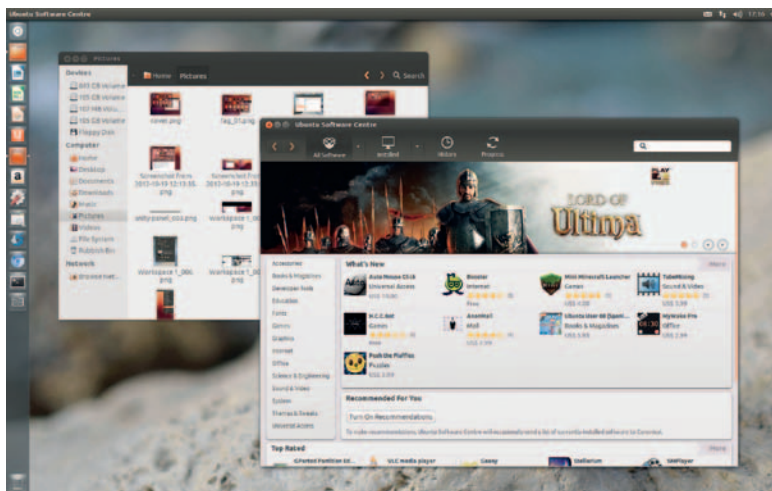
Грэм Моррисон приручает Квантильного Кетсаля, недоумевая, уместно ли было давать ему сперва одичать.

Вкратце

» Ubuntu — один из самых популярных дистрибутивов Linux. Он щеголяет собственным рабочим столом Unity и большим онлайн-сообществом. См. также: производный от Ubuntu Linux Mint.

Пару лет назад каждый новый выпуск Ubuntu был большим праздником, по крайней мере здесь, в Башнях LXF. Разработка шла от успеха к успеху, Ubuntu становилась чем-то вроде негласного стандарта Linux для всех. Она быстро набирала известность в нетехнических кругах, давая людям понять, что Linux бывает и с человеческим лицом, а не только с командной строкой. Сияние успехов Ubuntu озаряло всех нас. Но вдруг путь Ubuntu стал темнеть, сбиваясь на бездорожье в поисках приключений. Заменяв свободный рабочий стол своей собственной разработкой, внедряя в дистрибутив все больше платных сервисов, Ubuntu фактически стал одним из самых коммерциализованных дистрибутивов, отвращая традиционалистов Linux, да и не выполняя своих обещаний по легкости использования.

С последним выпуском, в апреле, мы впервые подумали, что этот план может дать положительные результаты, сделав своего преемника — Ubuntu 12.10 — пер-



» Обзаведясь новыми «линзами» и HUD, Ubuntu взаправду стал мечтой опытного пользователя. Но сложность идет рука об руку с утратой стабильности.

«Ubuntu 12.10 — первая Ubuntu, которой мы не опасались.»

вой за недавнее время Ubuntu, которой мы не опасались. Полные этой новой уверенности, мы попытались обновить нашу установку 12.04, не приступая к установке заново. В LTS-выпуске для этого

нужно разрешить обновления, не помещенные как «с долгосрочной поддержкой». Это было ошибкой. Скачав больше гигабайта дополнительных пакетов и потратив 90 минут на их установку, после перезагрузки Ubuntu сломалась окончательно.

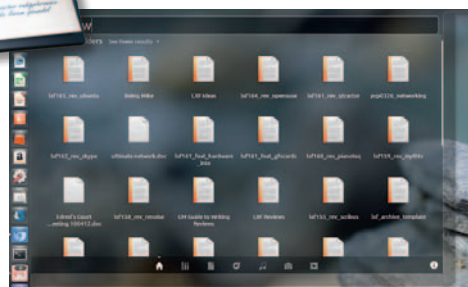
Ускоренный открытый драйвер Nouveau для видеокарт nVidia мигал экраном, а рабочий стол Unity рухнул через 10 секунд после загрузки. Потратив 20 минут на решение проблем этой установки, мы решили плюнуть на обновление и поставить все с нуля. Печально, что нельзя рекомендовать обновление на месте. Но, возможно, это и не обновление сломало нашу систему, а сама Ubuntu.

Плохое

Свежая установка заработала без проблем, но новый рабочий стол по-прежнему не сиял стабильностью. Драйвера Nouveau работали лучше, чем при обновлении, но низкая производительность принудила нас поставить несвободные драйвера, для чего пришлось активировать дополнительные драйвера, спрятанные в программе Software Sources [Источники приложений]. Увы, проблемы производительности не исчезли: и стабильность, и скорость оставляли желать лучшего.

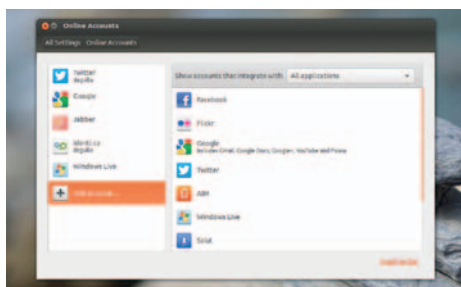
От экранных уведомлений оставалась только тень, когда мы пытались по ним щелкнуть; а меню, переехавшие с каких-то web-страниц в верхнюю панель, работали не всегда. Compiz тоже норовил падать, увлекая за собой рабочий стол. Тема оформления по умолчанию в программе обмена сообщениями Empathy отказывалась загружаться без перенастройки, а функция блокировки экрана перед запросом пароля заставляла нас секунд пять в прострации молотить по компьютеру. Проблемы могут быть вызваны специфической работой Compiz с некоторыми функциями OpenGL; и хотя над ними уже работают, исправление не было доступно к окончанию этого обзора. Такие сложности и серьезные проблемы производительности с железом Nvidia могут быть критичны. Мы можем только надеяться, что на момент вашего чтения они уже устарели. Установка с графической картой

Свойства навскидку



Поиск по Google Docs

Нам понравилось, что документы Google Docs можно искать прямо из Dash, хотя открываются они по-прежнему в браузере.



Объединенные учетные записи

Все ваши учетные записи онлайн можно соединить в одном апплете, настраиваемом в одном окне.

Intel прошла лучше, но производительность виртуализации была ужасна.

Уродливое

Нам также не понравилось, что продукты Amazon сейчас засоряют результаты поиска. Особенно это раздражает, когда ты только установил систему и хочешь найти системные утилиты, нужные для получения оптимальной конфигурации. Хуже того, результаты могут быть непристойными. Никаких семейных фильтров здесь нет (посмотрите первые 15 результатов по слову “hard [жесткий]”, если не верите), что наводит на мысль: Ubuntu – не та система, которую хочется поставить своим детям. Добавим сюда шум, рассеяние внимания и нарушение приватности – и оценкой «опции» Amazon будет «полный провал». Можно отключить любой онлайн-поиск с панели настроек приватности, но это обернется отключением и полезных онлайн-опций, таких как поиск по Google Docs. Мы отскели опцию, удалив пакет *unity-lens-shopping*, но чтобы результаты поиска в Amazon стали приемлемыми, их надо фильтровать.

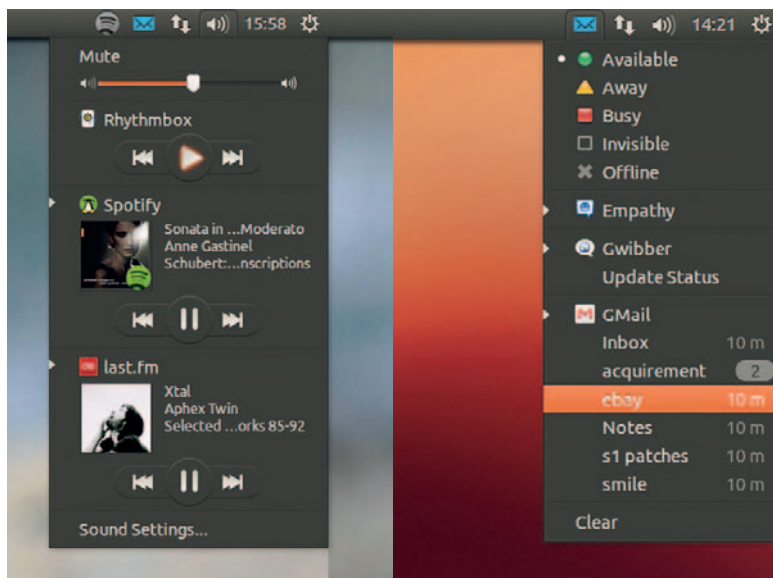
Качество локального поиска Ubuntu тоже под вопросом. Например, файл данной статьи назывался *lxf165_rev_ubuntu*, но был показан в самом низу результатов поиска по словам и ‘lxf’, и ‘ubuntu’. Кроме того, значки онлайн-документов не отличаются от их локальных аналогов; щелкнув по файлу, вы можете неожиданно угодить в окно браузера. Похоже, этот поисковый механизм – наследие ужасного поиска в Ubuntu Software Centre. Нет приоритетов для поисковых слов, обнаруженных в имени документа, по сравнению с нахождением в самом документе; не выделяются недавно открытые документы. Если намерение Canonical – потеснить Google собственным поиском, результаты должны быть получше.

Хорошее

Многие новые функции нам пришлось по душе. Вы можете скрыть съемные носители из меню запуска, выбрав в контекст-

» **Использование в Ubuntu унифицированных меню для учетных записей онлайн и аудио может стать одним из лучших ее свойств.**

» **Нам понравилось включение результатов iPlayer; возможно, наше недовольство Amazon связано с неуплатой им налога в £7 млрд с продаж в Великобритании.**



ном меню Unlock from Launcher, а объединенное меню сообщений выглядит отлично. Нам понравилось, что столько онлайн-инструментов можно настроить в одном окне и управлять ими из одного меню. Но хотелось бы некой приоритетности сообщений, позволяющей видеть, что кто-то хочет пообщаться, на шумовом фоне постов из Twitter.

WebApps в этом выпуске также обновились. При посещении совместимого с WebApp сайта – например, BBC News, Google Mail/Docs/Reader или eBay – Ubuntu оформит данные со страницы в виде псевдо-приложения. *GMail* становится полноценным почтовым клиентом, со значком «непрочитанные сообщения» и дополнительными пунктами в меню Communications. *Last.fm* управляется ползунком громкости, а свежие обновления отображаются в экранных уведомлениях. Все эти функции работают в *Firefox* и *Chrome*, но для последнего нужно специально установить модуль расширения. Кстати, простого способа отключить установленные приложения нет. *Chrome* с *Last.fm* пару раз у нас падали, но в целом WebApps – действительно инновация, и нас порадовало то, как далеко продвинулась Ubuntu в этом направлении.

Также Ubuntu – лидер дистрибутивов по части аудио. Звук работает безупречно, доказывая, что *PulseAudio* вообще-то не так уж плох. Новые звуковые приложения коммутируют входы и выходы без вмешательства пользователя, регулировка громкости через апплет работает блестяще, как и возможность переключения между устройствами – и все это без малейших накладок, отказов и перезагрузок.

Но в целом создается ощущение, что отдельные части системы разрабатываются в разном темпе; а итог – недоста-

ток надежности, баланса и слаженности. При всей красоте новых функций, хотелось бы видеть лучшие значки по умолчанию и больше настроек Launcher-A. Если *Compiz* и *Unity* собираются стать приметой рабочего стола, они должны работать безупречно. Это означает лучшее тестирование, меньше системных сбоев и лучшую производительность. Унифицированные меню приложений также не должны быть проблемой; новые пользователи, запустившие *GIMP*, будут ошарашены, когда они выберут панель инструментов и ничего не появится.

Это сложные проблемы, но Ubuntu вступила на этот путь добровольно. Для продвижения смелых идей, таких как коммерческое партнерство и унифицированные меню, нужны серьезные усилия. И хотя трудно не любить Ubuntu за ее дерзкие попытки отличаться от всех, переход становится болезненным. Ubuntu нужно либо наворачивать и быстро учиться, либо перейти на выпуск только LTS-релизов; и ни один из этих вариантов нам не нравится. **LXF**

LINUX FORMAT **Вердикт**

Ubuntu 12.10

Разработчик: Canonical Ltd
 Сайт: www.ubuntu.com
 Лицензия: GPL

Функциональность	8/10
Производительность	4/10
Удобство использования	7/10
Документация	7/10

» Ubuntu богата идеями, но бедна усилиями заставить их работать как задумано.

Рейтинг 6/10



Slackware 14

Старейший в мире дистрибутив предпочитает простой и открытый дизайн колышущимся окнам, объясняет **Майк Сондерс**.

Вкратце

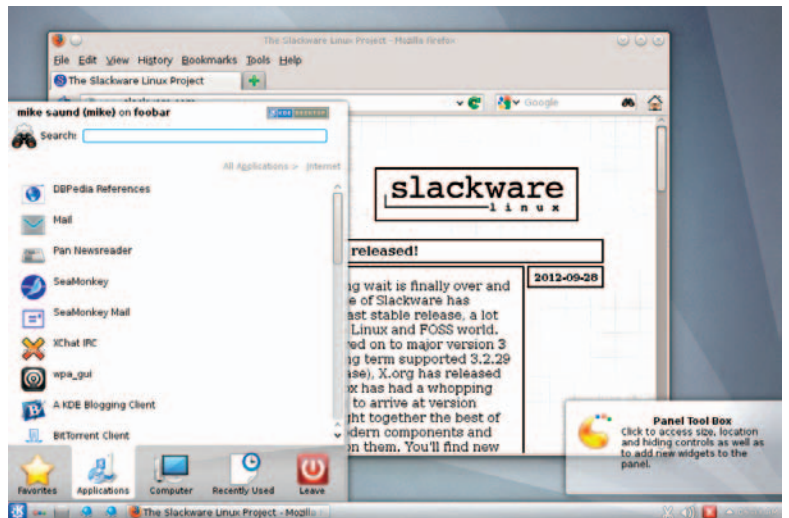
» Классический дистрибутив Linux, созданный большей частью одним разработчиком и не основанный ни на каком другом. См. также: VectorLinux и Zenwalk (оба — потомки Slackware).

Вы вряд ли увидите на интернет-форумах жалобу «Последнее обновление Slackware полностью сломало мою систему!» (Сравните это с Ubuntu и Fedora, где неумовимые изменения «под капотом» между выпусками нередко оборачиваются лавиной отказов драйверов, заставляя пользователей сидеть без звука или тупо паяльниками в подкачку командной строки. Мы видим такое слишком даже часто.)

Здесь вы могли бы возразить: мол, все дело в сообществе пользователей Slackware; ведь большинство «слакеров» — технари. Но кроме того, это самый консервативный дистрибутив Linux — с 1993 г. он работает себе без всяких фанфар, верный своей первоначальной цели: быть простым и стабильным дистрибутивом общего назначения. Его основной (а большую часть времени и единственный) разработчик Патрик Фолькердинг [Patrick Volkerding] — спокойная и неприязнительная фигура в Linux-сообществе. Он не ищет славы и почета; он никогда внезапно не объявит, что Slackware отныне ориентируется на планшеты или нечто в том же духе.

Именно этот подход одиночки столь притягателен для многих давних линуксоидов. Вы знаете, чего ждать. Slackware не стремится выскочить к рампе и завоевать настольные ПК, как Ubuntu, и не погружает в политику и разборки, как Debian. Это всего лишь один ну очень толковый парень — и несколько помощников — собирающий вместе свежие компоненты Linux и популярные открытые приложения.

Дистрибутив, который вы получаете в итоге, максимально чист: пакеты с программами почти идентичны их оригинальным версиям, никаких сотен дистрибутивных



» Хорошая новость для KDE-пуристов: Slackware не горюит ничего вокруг оригинальных исходников, оставляя ваш рабочий стол в чистейшем виде.

символических доработок. Пакеты — это обычные архивы исходников плюс немного метаданных, а пакетная система очень проста. Вы, возможно, сталкивались с жуткими сценариями в других дистрибутивах, когда через змеиный клубок зависимостей удаление экранной заставки норовит заодно уничтожить и стартовые скрипты. Здесь такое исключено.

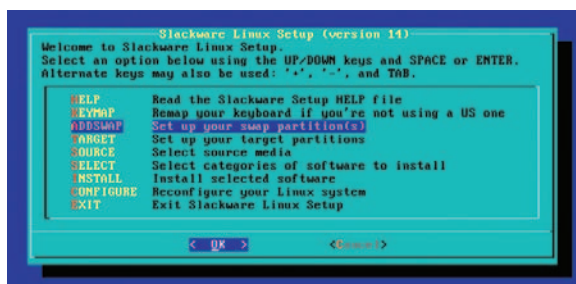
Что нового в версии 14?

Установка, как и в предыдущих выпусках, требует некоторых базовых знаний Linux (например, разметить диск вам придется с помощью `cfdisk`). Сам установщик минималистичный и текстовый. Поклонники Slackware известны своим специфичным юмором, который и выплеснется на вас на этапе выбора пакетов: вы можете либо установить все, что есть на дисках, либо вручную выбирать пакеты по одному, что в случае установки с иксами «займет примерно год».

В итоге вы получите стандартный KDE 4.8.5 (альтернатива — `Xfce 4.10`) вместе с широким выбором приложений, включая `Firefox 15`, `GIMP 2.8` и офисный пакет `Calligra 2.4` [бывший `Koffice`, — прим. пер.]. Самые заметные бреши — `Gnome` и `LibreOffice`, доступные в виде сторонних пакетов в других источниках. (`Gnome` был отброшен в 2005 г. из-за сложностей с пакетированием.) Все это базируется на ядре `3.2.29`, `glibc 2.15` и `gcc 4.7.1`. Что касается

документации, 14-я версия замечательной книги `Slackbook` на момент написания статьи была еще в состоянии бета-версии.

Основное преимущество (и недостаток) Slackware — это главный разработчик. Уловив стиль работы Патрика, вы оцените стройность и простоту дистрибутива. Мы бы не советовали применять Slackware в бизнесе, ведь если Патрик бросит заниматься дистрибутивом, никто не знает, когда выйдут (и выйдут ли) следующие обновления безопасности. Но если вы ищете дистрибутив, в котором прозрачный дизайн важнее эффектов прозрачных окон, Slackware — хороший выбор. **LXF**



» Инсталлятор Slackware, хоть он и текстовый, позволит провести установку большей частью нажатиями `Enter`.

LINUX Вердикт
FORMAT

Slackware 14

Разработчик: Патрик Фолькердинг
 Сайт: www.slackware.com
 Цена: Бесплатно для скачивания, платная коробочная версия

Функциональность	6/10
Производительность	9/10
Удобство использования	7/10
Документация	8/10

» Просто, чисто и надежно. Не введет вас в экстаз, но и не выбьет почву из-под ног.

Рейтинг 8/10

ROSA Server 2012

Бен Эверард погружается в новый серверный дистрибутив ROSA и изучает, справилась ли российская компания с задачей.

Вкратце

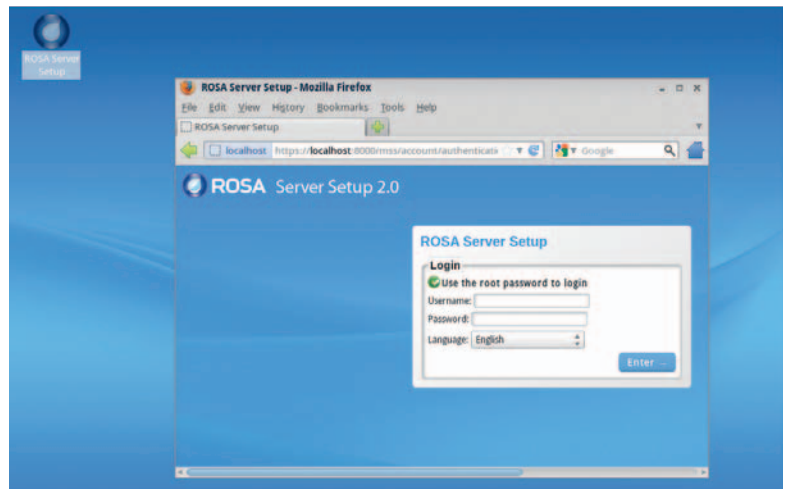
» Клон RHEL, цель которого — упростить администрирование серверов. См. также: CentOS, Scientific Linux и Ubuntu Server.

РОСА — относительный новичок на Linux-сцене. Российская компания выпустила свой настольный дистрибутив — ответвление от Mandriva — в конце 2011 г. Он привлек внимание своим вариантом рабочего стола KDE и впечатлил нас настолько, что мы включили последнюю версию на наш диск. Год спустя компания выпустила серверный дистрибутив. Это не просто модификация настольного дистрибутива, а заново построенная система на базе кода Red Hat Enterprise Linux.

Построение настольного и серверного дистрибутивов на разных кодовых базах — ход дерзкий, но не безрассудный. Он показал: компания понимает, что две области применения обладают разными требованиями, и не ограничивается поверхностными изменениями.

Установку, которая проходит без проблем, любезно выполняет Anaconda от Red Hat. Надо выбрать, какой тип сервера установить, но все остальное делается автоматически. Первое, что вы замечаете при загрузке дистрибутива — та самая версия KDE, впечатлившая нас в настольной версии, ушла: на сервере это было бы бесполезной тратой ресурсов процессора. На ее месте — LXDE, в тех же синем и сером цветах; но кто же выберет серверный дистрибутив для десктопа.

Уникальный козырь данного дистрибутива — *ROSA Server Setup*, локально настроенное web-приложение, помогающее пользователю настроить различные службы. Пока доступна настройка только нескольких служб, но они покрывают большинство областей, включая web, базы



» Из России с любовью... и с интуитивно понятным web-интерфейсом.

данных, файлы, LDAP и другие. Если у вас нет особых требований, вы найдете здесь все, что вам нужно. Правда, какого-то конкретного сервера может и не оказаться.

Это касается не только web-интерфейса. Заглянув в установку/удаление программ, мы обнаружим, что пакетов там тоже меньше обычного. Подозреваем, что разработчики поступили так ради обеспечения качества немногих доступных вариантов, и по нашему мнению, это хорошая идея, но вы должны понимать, что некоторые вещи, которые вы рассчитываете увидеть, могут отсутствовать.

Хороший старт

Web-интерфейс запускается по умолчанию, что подразумевает возможность после установки управлять всем дистанционно, но мы обнаружили, что ROSA Server не смогла поднять сетевое соединение.

Управление пакетами не дополнено другими логичными инструментами управления. К примеру, установив *Apache*, придется вернуться в командную строку, чтобы запустить его. Это несложная задача для любого, кто работает с серверами, но вполне достаточно, чтобы сбить с толку неопытного пользователя.

Нам понравилось средство установки программ. Если бы интерфейс настройки сервера был дополнен хотя бы базовыми опциями администрирования, такими как запуск и остановка сервисов, это был бы отличный дистрибутив для не слишком

искушенных пользователей, которым нужен легкий в настройке сервер. Конечно, это лишь первый выпуск дистрибутива; возможно, мы хотим от него слишком много. Но того, что мы увидели, достаточно, чтобы с нетерпением ожидать следующих версий продукта.

Тот факт, что он основан на RHEL, означает, что дистрибутив будет надежен как скала и совместим почти с любым серверным ПО. Но если именно этого вы ищете, на данном этапе мы рекомендуем более развитые клоны RHEL, такие как Scientific Linux или CentOS, исходя из их происхождения. Это изменится в следующих версиях, если ROSA Server хорошо покажет себя в реальных тестах. **LXF**

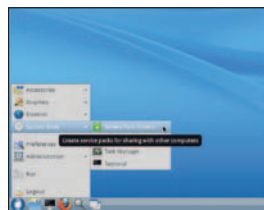


Свойства навскидку



Настройка

ROSA Server Setup может вырасти в очень полезный инструмент, но пока довольно ограничен.



Рабочий стол

LXDE — превосходный выбор для легковесного настольного сервера.

LINUX Вердикт
FORMAT

ROSA Server

Разработчик: ROSA Labs
 Сайт: Rosalab.com
 Цена: Бесплатно

Функциональность	6/10
Производительность	8/10
Удобство использования	9/10
Документация	6/10

» Хороший дебют серверного дистрибутива; с нетерпением ждем следующего релиза.

Рейтинг **8/10**

Raspberry Pi 512MB

Чтобы удвоить память на своем Amiga 500, Грэм Моррисон заплатил целое состояние. А вот это обновление – совершенно бесплатное.

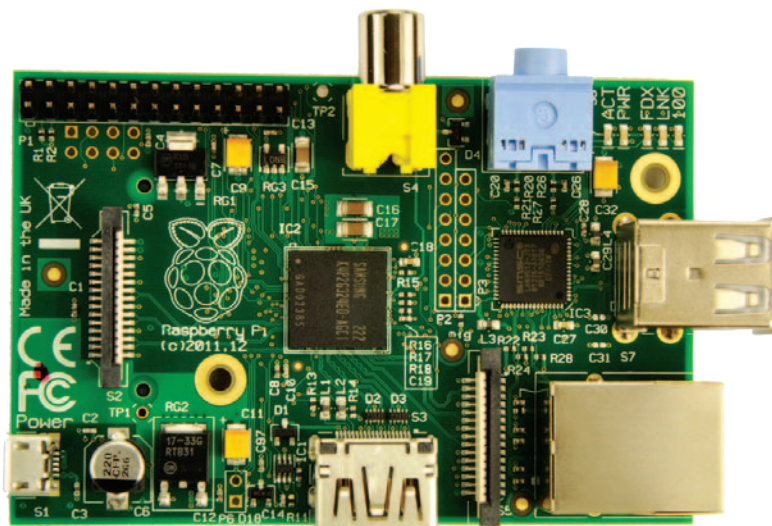
Вкратце

» Сверхдешевый мини-компьютер на ARM, способный запускать многие разновидности Linux. См. также: SheevaPlug.



Скромный Raspberry Pi стал известен всего лишь как прототип в конце 2011 г. Хотя он был в разработке уже 4 года. Это детище Эбена Аптона [Eben Upton], который начал собирать Pi на самой длинной макетной плате, которую смог купить в магазине электроники Marlin. Основным стимулом к его созданию было желание сделать компьютерные науки общедоступными; планировалось дать эти платы школьникам в надежде, что они загорятся энтузиазмом к хакерству, как основатели Raspberry Pi с их домашними компьютерами в 1980-е.

Этот компьютерный мем 1980-х, вместе с адресной книгой еще одного «заговорщика», Дэвида Брейбена [David Braben], и идей, что программированию нужно обучать всех, помог феноменальному продвижению Raspberry Pi. Всего через несколько месяцев после демонстрации прототипов более 200 000 человек заре-



» Raspberry Pi Model B теперь производится в Уэльсе и оснащен крепежными отверстиями, большим количеством GPIO и двойным объемом памяти за ту же цену.

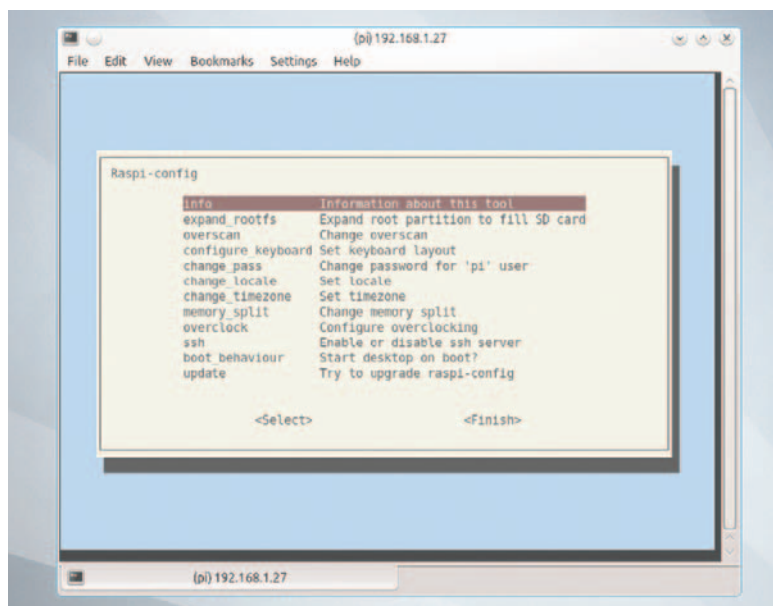
«Главное достоинство Pi — не спецификация железа, а цена.»

гистрировались для получения плат модели B, что в 20 раз превысило первую партию выпуска. За 6 месяцев пестрая

демографическая группа из хакеров, мастеров и самоделкиных скупила пол-миллиона штук. Это устройство можно назвать полноценным ПК, втиснутым в миниатюрную печатную плату размером всего лишь 85,60 на 53,98 мм. Процессор не особо мощный – ARMv6, работающий на стандартной частоте 700 МГц; но устройство может питаться от USB-кабеля и даже от батареек и способно запускать графическое окружение с ускоренным видео как через HDMI, так и через разъем

RCA, подключаемый к телевизору. Не хватает старого доброго VGA, ведь сейчас так много старых неиспользуемых мониторов; но это скорее ограничение используемого чипа SoC, чем дизайна устройства. А главное достоинство Pi – не спецификации железа, а цена. Он может стать вашим, с доставкой, всего за \$35, включая НДС. Спрос на него столь велик, что более дешевая бессетевая модель A все еще не выпущена. Но вскоре после выхода модели B были выпущены три ее ревизии. Первые две, помеченные как 2 и 3 в идентификаторе процессора, отличаются лишь отсутствием компонента D14, изначально расположенного под портом HDMI, и некоторыми изменениями в разводке платы. Но третья обновление, официально названное Revision 2.0 и доступное с конца сентября, включает более серьезные изменения, хотя и выглядит почти так же, как предыдущие версии. «Почти» относится большей частью к двум дополнительным монтажным отверстиям для облегчения крепления к корпусу и добавленному разъему P5, состоящему из питания, земли и четырех дополнительных соединений для портов GPIO.

» Вроде и немного, но возможность управления распределением памяти между GPU и CPU делает это устройство чрезвычайно гибким.



Двойное счастье

Самое заметное улучшение – это удвоение объема памяти с 256 до 512 МБ при сохранении прежней цены, что делает Raspberry Pi более пригодным для запуска

настольных приложений. На нашей системе стандартная загрузка с командной строкой оставила свободными 384,4 МБ – куда лучше, чем 126 МБ на предыдущей модели. Контакты GPIO также были доработаны, включая предназначенные для широко используемой шины данных I2C. В этой области свойства Raspberry Pi пересекаются с другими проектами, подобными Arduino: это возможность создавать собственное оборудование и подключать его как через цифровой, так и аналоговый сигнал, управляемый ПО Raspberry Pi. Несомненное преимущество Pi в возможности запускать на нем полноценную ОС, а не байт-код.

Raspbian Pi

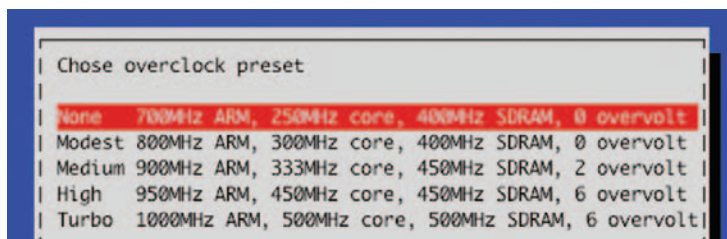
Официальная ОС – производный от Debian дистрибутив Linux под названием Raspbian. Он также немного изменен, и хотя мы смогли успешно установить систему на SD-карту 2 Гб, удобнее будет взять носитель большего объема – это дает больше свободы в добавлении собственных файлов. Если вы уже работали в командной строке дистрибутивов на базе Debian, у вас не будет никаких проблем: многие из команд и опций окажутся знакомыми. Но даже если вы использовали только Windows, командная строка не так уж сложна. Вооружитесь парой руководств и приступайте к исследованиям. Если вы сломаете установленный дистрибутив, то худшее, что может случиться – придется переписать носитель свежей версией дистрибутива.

«Если Linux вам внове, учиться использовать Pi будет сложно.»

После первой загрузки вас попросят запустить утилиту настройки *raspi-config*, недавно обновленную для Revision 2.0. Этот инструмент – одно из главных преимуществ использования официального дистрибутива, а не любого другого доступного. В программе настройки вы сможете увеличить размер файловой системы на флешке, изменить раскладку клавиатуры и пароль, а также настроить запуск графического окружения при загрузке. Дополнительные опции позволяют настроить распределение памяти между центральным и графическим процессором, что прекрасно, если вы хотите поупражняться с играми или предоставить максимум ресурсов приложениям.

В этом релизе вы также можете разогнать Pi (см. врезку вверху). В наших тестах файл в 1 Гб из */dev/urandom* был создан за 151 секунду без разгона и за 118

Разгон



Теперь вы можете официально разгонять новый Raspberry Pi – утилита *raspi-config* позволит увеличить частоты процессора (700 МГц), ядра (250 МГц) и памяти (400 МГц) до 1000 МГц, 500 МГц и 500 МГц соответственно. Для этого может потребоваться более мощный источник питания, и возможны проблемы с отводом тепла. Но у нас все это сбавало с обычным USB-хабом.

Разгон динамический, т.е. новые частоты будут включаться только при необходимости, а при температуре ядра примерно 52 градуса скорость будет сброшена до обычной. Вы можете отследить это в реальном времени, наблюдая за значениями `thermal_zone0/temp` и `caling_cur_freq` на виртуальной файловой системе `sys`.

при 1 ГГц. Сжатие этого файла *bzip2* заняло 439 секунд при 700 МГц и 333 секунды при максимальном значении разгона. Это говорит о примерно 30 % прироста производительности; результаты можно легко улучшить при лучшем охлаждении. Но как всегда, тесты далеки от реальной работы, и ваши результаты могут отличаться.

Пи в квадрате

Как применить Raspberry Pi, полностью зависит от вас. Благодаря использованию Debian вы можете устанавливать в точности те же программы, что и в любом другом дистрибутиве Linux. Если вам нужен доступ к контактам GPIO, для этого есть утилита командной строки и несколько API для различных языков. Но поскольку Raspberry Pi уже угнездился в сообществе хакеров, новые возможности появятся очень скоро, включая управление через web-интерфейс и несметное количество других проектов. Возможности программирования легко удовлетворяются установкой чего угодно, от Python до недавно анонсированного TinyBASIC, если вы ищете подлинного опыта 1980-х.

Существует бурно развивающееся онлайн-сообщество, а поточно пополняющиеся wiki предоставляют всю необходимую информацию для новичков (хотя было бы неплохо собрать все это в простейшее руководство). Новые устройства изготавливаются на производственных мощностях Sony в Уэльсе, причем создан важный прецедент выпуска в сотрудничестве с Broadcom частей видеоядра под открытой лицензией.

Есть у Raspberry Pi и свои проблемы. Используется процессор старой технологии – он не так быстр, как его конкуренты. Отсутствие корпуса может ограничить

применение; хотя существуют корпуса от сторонних производителей. И если Linux вам внове, учиться полноценно использовать Pi будет сложно, поскольку придется записывать образ на карту памяти, обновлять пакеты и лезть в командную строку для того, чтобы хотя бы добраться до графического интерфейса. Но все это – часть сюжета, и облегчение таких задач ограничило бы гибкость устройства.

Эта гибкость изумительна, но она бы мало что значила без низкой цены. Сделав Raspberry Pi столь дешевым, его создатели дали новому поколению возможность экспериментировать, не боясь испортить дорогое оборудование. И это ключ к успеху Pi. Он стирает десятилетия удушающего формализма, традиционно окружающего микрокомпьютерные науки, заменяя его миром возможностей. Согласитесь: даже у вас уже возникла идея, что именно можно попробовать сделать с Raspberry Pi; и это причина его нынешнего и будущего успеха. **LXF**

LINUX **Вердикт**
FORMAT

Raspberry Pi 512MB

Разработчик: Raspberry Pi Foundation
Сайт: www.raspberrypi.org
Лицензия: £25 (RS Components)

Функциональность	10/10
Производительность	8/10
Удобство использования	8/10
Оправданность цены	10/10

» Дешевый и маленький, он определенно изменит мир.

Рейтинг 9/10



НОВАЯ МОБИЛЬНАЯ ПЛАТФОРМА

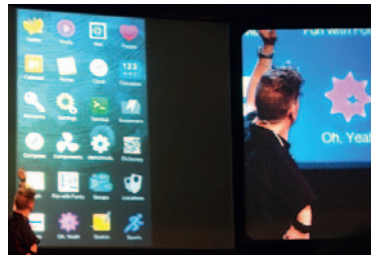
MeeGo бросает вызов Android

Компания Jolla представила основанную на MeeGo операционную систему Sailfish.

Android в ушедшем году завоевал почти 60 % рынка. И люди пытаются повторить успех Google. Одна из таких компаний, Jolla, основанная бывшими сотрудниками Nokia с целью разработки новых смартфонов на основе Linux-платформы MeeGo, представила новую ОС Sailfish, анонсировала SDK для разработки приложений и продемонстрировала возможности первого смартфона на ее основе. Кроме того, компания вводит программу по лицензированию использования Sailfish на устройствах сторонних производителей, желающих выпускать аппараты на базе новой ОС.

Изначально планировалось сформировать мобильную экосистему на китайском рынке, и только потом начинать продвижение на европейский и американский рынки. Тем не менее, интерес к платформе среди европейских компаний оказался достаточно высок, и уже заключено соглашение по поставке телефонов Jolla с DNA, третьим по величине сотовым оператором Финляндии. Также налажено сотрудничество с компанией ST-Ericsson, в рам-

» Новая мобильная ОС сильно напоминает Android — как внешне, так и внутренне. Особенно внешне.



ках которого уже обеспечена поддержка в Sailfish устройств на базе аппаратной платформы NovaThor ModAp. В будущем компания Jolla намерена подготовить редакцию Sailfish для планшетов, умных телевизоров и автомобильных информационно-развлекательных систем.

Для разработки приложений для устройств на базе Sailfish предлагается использовать фреймворк Qt. ОС Sailfish основана на наработках открытых проектов, среди которых MeeGo, Qt и Mer Core. Код базовой части Sailfish планируется распространять под открытой лицензией, а все вносимые изменения возвращать в родительские проекты. Из оригинальных

наработок — новый пользовательский интерфейс на базе Qt, кардинально отличающийся от используемого в смартфоне N9 варианта MeeGo. По сути Sailfish представляет собой базовые системные компоненты Mer Core, поверх которых запускается Qt-интерфейс пользователя, построенный на технологиях QML и QtQuick. Окружение Sailfish будет совместимо с приложениями, развиваемыми в рамках дистрибутива Nemo, разработанного силами участников сообщества MeeGo. В платформу планируется интегрировать прослойки для выполнения приложений, написанных для платформ Android и Tizen. Получается, что в будущем Android может превратиться в универсальную систему: ОС, среда разработки и прослойка совместимости. Хорошо это или плохо — покажет время. Пока можно лишь отметить и без того излишнюю фрагментированность кода Android.

Что касается MeeGo и Sailfish — ими могло бы быть укомплектовано все многообразие смартфонов Nokia, не выбери эта финская компания сотрудничество с Microsoft.

ИНСТРУМЕНТАРИЙ ДЛЯ УМНИКОВ

Новая версия Android-SDK

Больше нельзя создавать производные продукты, нарушая совместимость.

Компания Google опубликовала новую версию SDK для платформы Android, а также связанных с ним инструментария Android SDK Tools 21, ADT Plugin 21 для Eclipse и пакета Android NDK 8c (Native Development Kit), позволяющего организовать выполнение «родного» кода в окружении Android.

Реализована поддержка особенностей нового Android 4.2 — например, добавлены возможности запуска Renderscript-вычислений прямо на GPU, вывода отдельного контента на дополнительные экраны (ранее было доступно только зеркалирование основного экрана), создания интерактивных виджетов.

Новая версия пакета SDK Tools примечательна реализацией нового визуального редактора меню, дополнительными правилами Lint, новыми шаблонами для быстрой разработки приложений, новым фреймворком для автоматического тестирования интерфейса пользователя.

Лицензионное соглашение по SDK стало более жестким, с целью защиты платформы Android от фрагментации, то есть появления под именем Android сторонних реализаций, не совместимых с базовой платформой от Google. Соглашение теперь явно запрещает создание и распространение сторонних вариантов SDK, предоставляющих нестандартные API,

которые не определены в официальных спецификациях.

Ранее производители могли добавлять в прошивку дополнительный API и поставлять отдельный SDK, генерирующий сборки, совместимые только с его вариантом платформы.

Участники альянса Open Handset Alliance (OHA), курирующего разработку мобильной платформы Android и включающего в свой состав большинство крупных производителей Android-устройств, обязались не создавать несовместимые продукты, но на сторонние компании до сих пор данные ограничения не распространялись, что стало поводом для злоупотреблений.

ДВА ЭКРАНА ЗА ДВА ЧЕРВОНЦА

Идея реализована через два года

Компания Yota готова показать миру новый двухэкранный смартфон.

Компания Yota Devices разработала новый смартфон с двумя экранами на основе комплектующих из Азии. Yota Devices, выделившаяся из «Скартела» в 2011 году, намерена представить его на Mobile World Congress в Барселоне в феврале 2013 года.

Итак, идея «первого русского смартфона» от Yota, прототип которого демонстрировал тогдашнему президенту Медведеву глава госхолдинга «Ростехнологии» Сергей Чemezov осенью 2010 года, жива. Тогда главе государства рассказали, что «Скартел» сам занимается разработкой на базе ОС Android, дизайн устройства создан «очень талантливым выходцем с Украины», а партнер, который будет выбран в ходе тендера, займется лишь его производством. Было обещано, что смартфон появится в продаже в середине 2011 года.

Тогдашний «прототип» при ближайшем рассмотрении оказался куском пластмассы с наклейками, но аппарат, который Yota Devices повезет в Барселону – уже полноценное работающее устройство.



› Лед тронулся — новые смартфоны на Android от Yota скоро появятся в магазинах России.

Как поведал прессе гендиректор компании Владислав Мартынов, дизайн и ПО на основе Android – плоды двухлетних трудов российской команды Yota Devices.

Помимо GSM и 3G, устройство поддерживает стандарт мобильной связи четвертого поколения LTE, в котором и работает Yota, и снабжено 12-мегапиксельной камерой. Когда Медведеву гордо демонстрировали кусок пластмассы, Yota еще работала в стандарте WiMAX.

Главным козырем изделия в Yota считают наличие на задней панели смартфона второго, черно-белого дисплея (диагональ – 4,3 дюйма), выполненного по технологии электронных чернил (e-ink), используемой в современных букридерах. На него можно выводить SMS, электронные письма и книги, и т.п. Монохромный экран потребляет меньше энергии и сохраняет изображение постоянно, разгружая основной цветной дисплей. «Вспо-

могательные» дисплеи для Yota Devices производит тайваньская компания E-ink, основной дисплей – Japan Display (совместное предприятие Toshiba, Sony и Hitachi). Микропроцессоры поставила Qualcomm. Контракт на сборку Yota Devices заключила, по словам Мартынова, с «крупным азиатским производителем, который собирает премиум-устройства для Apple и Google».

Розничная цена устройства в России предположительно составит около 20 тыс. рублей, при этом Yota мечтает договориться о продаже смартфона не только со «Скартелом», но и с рядом глобальных зарубежных операторов.

Попытки разрабатывать смартфоны в России или продавать китайскую продукцию под локальными марками предпринимаются регулярно, но россияне предпочитают дорогие смартфоны известных брендов, отметил аналитик Александр Венгранович. К тому же, по его словам, локальные разработчики редко способны привнести в свои аппараты ноу-хау, резко отличающие их от устройств-конкурентов.

ХАЛЯВЩИКИ В СЕТИ

Киберсквоттер решил нажиться

Один из участников проекта присвоил себе домен Cyanogenmod.com.

Разработчики независимой Android-прошивки CyanogenMod, 10-я версия которой вышла в конце ноября 2012 года, заявили о миграции проекта на домен cyanogenmod.org в связи с потерей контроля над прежним, еще со времен обитания разработчиков проекта на форумах XDA, доменом cyanogenmod.com. Недобросовестный бывший участник проекта, формальный владелец старого домена, выдвинул сообществу ультиматум, требуя выкупить у него домен за \$10 тыс.

За последние три года CyanogenMod превратился из хобби горстки энтузиастов в масштабный открытый проект, востребованный миллионами пользователей. Сообщество развивается на принципах взаимного доверия, и никто не подумал об официальной передаче проекту (а по сути, одному из разработчиков, поскольку за плечами CyanogenMod не стоит никакой организации) прав на домен.

В конце ноября владелец домена cyanogenmod.com, отвечающий за представительство проекта в Сети, дискредитировал себя попыткой обогатиться за счет сообщества через сделки по размещению платных ссылок на внешние сайты. Возмущенные этим, представители проекта потребовали передачи им прав на домен. Ответ был, что домен можно только купить...

В результате сайт вышел из-под контроля сообщества. Учетные записи конфликтера были удалены с представительств проекта в Twitter и Facebook, ему также был закрыт доступ к некоторым серверам в инфраструктуре проекта. В ответ шантажист заявил, что изменит данные в DNS и перенаправит домен на свой хост, и на подконтрольных ему системах удалил все учетные записи разработчиков CyanogenMod, заблокировал доступ к Google Apps и перенаправил на сторонний сервер электронные ящики @cyanogenmod.com.

В ICANN и в Google сообществом были отправлены запросы на возврат домена и восстановление доступа к Google Apps. Пользователям рекомендовали перейти на cyanogenmod.org.

Спустя сутки на сайте проекта появился анонс, сообщавший, что конфликт исчерпан и захватчик согласен безвозмездно передать cyanogenmod.com в руки сообщества. Тем не менее, сообщество приняло решение продолжить использование cyanogenmod.org в качестве первичного домена, организовав на cyanogenmod.com простое перенаправление.

Данное недоразумение продемонстрировало юридическую незащищенность открытой модели разработки – это серьезный урок для сообществ. Возможно, компании Google стоит задуматься над выделением специальной площадки для разработчиков альтернатив, коль скоро они существуют. **LXF**



Создаем объем с OpenGL



Наш эксперт

Джюльетта Кемп еще помнит времена, когда в фотоаппаратах была настоящая пленка. Сейчас она делает гораздо поболее снимков, чем тогда.

Джюльетта Кемп показывает, как «натянуть» экран телефона с Android на любезный многим вращающийся куб.

OpenGL (Open Graphics Library – открытая графическая библиотека) – широко используемая спецификация кроссплатформенного API для создания двумерной и трехмерной графики. Путем введения стандартного набора возможностей и интерфейса она упрощает работу с различными аппаратными платформами и 3D-ускорителями для разработчиков. Существуют привязки и реализации OpenGL для огромного количества языков программирования, и, к счастью, в Android API есть встроенная поддержка OpenGL. С первого релиза Android поддерживаются OpenGL ES 1.0 и 1.1, с Android 2.2 (API уровня 8) поддерживается OpenGL ES 2.0.

На этом уроке мы будем пользоваться OpenGL ES 2.0, так как сегодня почти на 85 % устройств используется Android 2.2 и выше, но если вам важна обратная совместимость с более старыми версиями, можно проверять уровень API в коде и при необходимости использовать OpenGL ES 1.1.

В этой статье я буду предполагать, что вы никогда не работали с OpenGL или с ее реализацией в Android API, и объясню некоторые основные идеи. Мы начнем со статичной двумерной фигуры, затем перейдем к вращающемуся кубу. На следующем уроке мы продолжим работу с этим примером, добавив в него работу с сенсорным экраном.

Начинаем знакомство с OpenGL

Поддержка OpenGL встроена в Android по умолчанию, и чтобы воспользоваться этим в своем проекте, ничего особенного делать не нужно. Однако надо добавить в манифест строку (поместите ее после **uses-sdk**), говорящую о том, что для запуска на устройстве приложение должно поддерживать OpenGL 2.0:

```
<uses-feature android:glEsVersion="0x00020000"
android:required="true" />
```

Основное Занятие [Activity] приложения не требует пояснений:

```
public class CubeInSpaceActivity extends Activity {
    private GLSurfaceView glView;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        glView = new GLSurfaceView(this);
        glView.setEGLContextClientVersion(2);
        glView.setRenderer(new CISGLRenderer());
        setContentView(glView);
    }
    @Override
    protected void onResume() {
        super.onResume();
        glView.onResume();
    }
    @Override
    protected void onPause() {
        super.onPause();
        glView.onPause();
    }
}
```

```
}
}
```

onCreate() создает используемый далее **GLSurfaceView** и контекст OpenGL ES 2.0 для него, и настраивает **Renderer**, который мы сейчас напишем. Для приостановки и возобновления приложения важно использовать методы Представления **onPause()** и **onResume()**. OpenGL великолепен, но потребляет много ресурсов, и нам незачем, чтобы он продолжал работать, пока пользователь не смотрит на нашу шикарную графику.

Большая часть работы выполняется классом **CISGLRenderer**; вот его первая версия, в ней задается только цвет фона:

```
public class CISGLRenderer implements GLSurfaceView.
Renderer {
    public void onSurfaceCreated(GL10 gl, EGLConfig config) {
        GLES20.glClearColor(61f/255, 89f/255, 171f/255, 1.0f);
    }
    public void onDrawFrame(GL10 gl) {
        GLES20.glClear(GLES20.GL_COLOR_BUFFER_BIT | GLES20.
GL_DEPTH_BUFFER_BIT);
    }
    public void onSurfaceChanged(GL10 gl, int width, int height) {
        GLES20.glViewport(0, 0, width, height);
    }
}
```

Для реализации **GLSurfaceView.Renderer**, нужно реализовать три следующих метода. **onSurfaceCreated()** вызывает-ся один раз при создании и задает параметры объекта; **onDraw-**

Frame() вызывается при каждой перерисовке кадра (постоянно); а **onSurfaceChanged()** вызывается при геометрических изменениях в Представлении (чаще всего при изменении ориентации экрана).

Обратите внимание, что у каждого из этих методов есть параметр **GL10**; так как мы пользуемся GLES20, использующим статические методы, этот параметр остается незадействованным (если вы хотите реализовать поддержку и для GL10, можете его задействовать).

glClearColor устанавливает цвет фона, который будет применяться каждый раз при вызове **glClear** – здесь это приятный кобальтово-синий. Параметры метода – красная (R), зеленая (G), синяя (B) составляющая и альфа (она управляет прозрачностью), каждый от 0 до 1. Для преобразования из стандартных 255 RGB можно, как и я, разделить стандартное значение на 255, или разделить вручную и подставить сюда результат.

Единственный аргумент **glClear()** – битовая маска очистки буфера: здесь это цвет и глубина (подробнее о буфере глубины позже, когда дойдем до трехмерных фигур).

Наконец, в методе **onSurfaceChanged()** мы задаем размеры прямоугольной области просмотра в соответствии с шириной и высотой поверхности Представления.

Но это только фон. Давайте определим квадрат, который нарисуем на нем, с помощью класса **Square**:

```
public class Square {
    private FloatBuffer squareBuffer;
    float vertices[] = {
        -0.5f, -0.5f, 0.0f, //bottom left
        0.5f, -0.5f, 0.0f, //bottom right
        -0.5f, 0.5f, 0.0f, //top left
        0.5f, 0.5f, 0.0f //top right
    };
    public Square() {
        ByteBuffer vbb = ByteBuffer.allocateDirect(vertices.length * 4);
        vbb.order(ByteOrder.nativeOrder());
        squareBuffer = vbb.asFloatBuffer();
        squareBuffer.put(vertices);
        squareBuffer.position(0);
    }
}
```

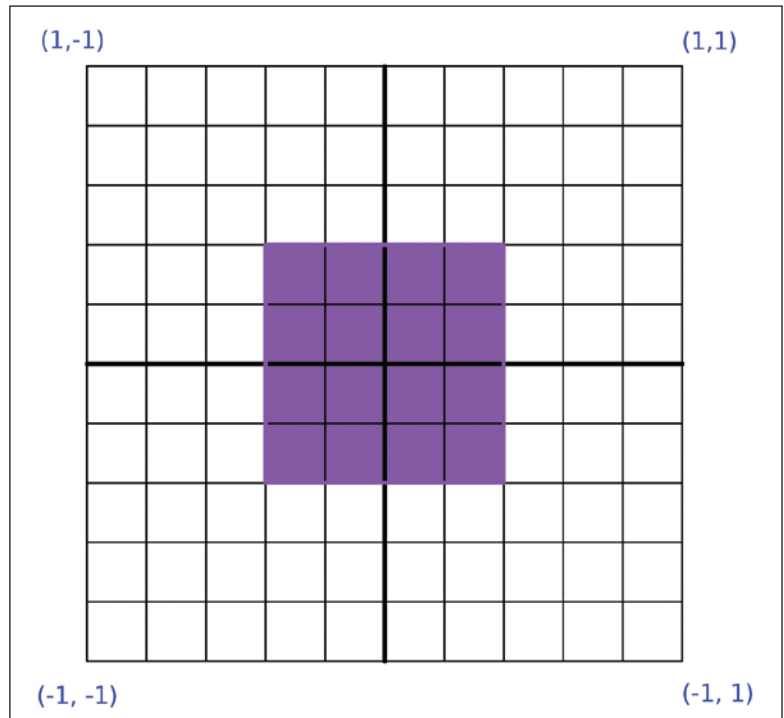
В массиве **vertices** задается расположение каждого угла (вершины) в координатах X, Y и Z (наш квадрат двумерный, поэтому координата Z везде равна нулю). Точка отсчета системы координат OpenGL находится в центре экрана, как показано на рисунке (Z положительные) и уходит вглубь экрана (Z отрицательные).

Создаем ByteBuffer

При задании параметров квадрата сначала создается байтовый буфер **ByteBuffer**, с помощью четырех байтов для каждой точки в обычном порядке байтов устройства. Затем с его помощью создается буфер с плавающей точкой **squareBuffer**; помещаем туда вершины и задаем начальную позицию на первой координате.

Вернемся в **CISGLRenderer.java**, чтобы настроить вершинный и фрагментный шейдеры. Вершинный шейдер управляет размещением и прорисовкой вершин (углов) фигур в OpenGL. Фрагментный шейдер управляет тем, что рисуется между вершинами. Для их создания нам понадобится объект **Strings**, который будет передан OpenGL:

```
private final String vertexShaderCode =
    "attribute vec4 vPosition; \n" +
    "void main(){ \n" +
    "    gl_Position = vPosition; \n" +
    "} \n";
private final String fragmentShaderCode =
```



► Двумерная система координат OpenGL. В трехмерной системе ось Z будет перпендикулярна экрану.

```
"precision medium float; \n" +
"void main(){ \n" +
    "    gl_FragColor = vec4 (0.63671875, 0.76953125, 0.22265625, 1.0); \n" +
    "} \n";
private Square square;
public void onSurfaceCreated(GL10 gl, EGLConfig config) {
    GLES20.glClearColor(0.5f, 0.5f, 0.5f, 1.0f);
    square = new Square();
    int vertexShader = loadShader(GLES20.GL_VERTEX_SHADER, vertexShaderCode);
    int fragmentShader = loadShader(GLES20.GL_FRAGMENT_SHADER, fragmentShaderCode);
}
private int loadShader(int type, String shaderCode) {
    int shader = GLES20.glCreateShader(type);
    GLES20.glShaderSource(shader, shaderCode);
    GLES20.glCompileShader(shader);
    return shader;
}
```

Чтобы создать шейдер, нужно загрузить код в виде строки **String** и скомпилировать его, как в методе **loadShader()**.

Для прорисовки в OpenGL ES 2.0 используется **Program**; это объект, который берет шейдеры (исполняемый код) и связывает их друг с другом, чтобы с их помощью нарисовать объекты. Поэтому нам нужно создать программу и связать с ней шейдеры:

```
private int program; private int positionHandle;
public void onSurfaceCreated(GL10 gl, EGLConfig config) {
    [...]
    program = GLES20.glCreateProgram();
    GLES20.glAttachShader(program, vertexShader);
    GLES20.glAttachShader(program, fragmentShader);
    GLES20.glLinkProgram(program);
    positionHandle = GLES20.glGetAttribLocation(program, "vPosition");
}
```

»

В этом коде создается новый пустой объект Program, с ним связываются оба наших шейдера, затем метод `glLinkProgram()` создает необходимый исполняемый код.

Наконец, в последней строке мы получаем дескриптор переменной `vPosition` в коде вершинного шейдера. Это позволяет нам передать расположение каждой вершины из нашего кода для Android в программу OpenGL.

Задав параметры программы OpenGL, добавьте следующие строки в метод `onDrawFrame()`, чтобы нарисовать квадрат:

```

GL ES20.glUseProgram(program);
square.draw(positionHandle);
а это __draw()__ method to __Square__:
public void draw(int positionHandle) {
    GL ES20.glVertexAttribPointer(positionHandle, 3, GL ES20.GL_
    FLOAT, false, 0, squareBuffer);
    GL ES20.glEnableVertexAttribArray(positionHandle);
    GL ES20.glDrawArrays(GL ES20.GL_TRIANGLE_STRIP, 0,
    vertices.length/3);
}

```

`glVertexAttribPointer()` определяет массив данных вершин. `positionHandle` – индекс отображаемой вершины, которая здесь состоит из трех компонентов (X, Y, Z). Тип данных – `GL_FLOAT`, значения не должны нормализоваться (`false`), между значениями массива вершин нет смещения (`0`), а информация о вершине находится в `squareBuffer`.

`glEnableVertexAttribArray()` активирует массив, а `glDrawArrays()` превращает массив данных в визуализированные геометрические примитивы. Мы пользуемся типом `GL_TRIANGLE_STRIP`, начиная индекс с нуля, и визуализируем четыре вершины, так как в массиве `vertices` для каждой вершины есть три значения (X, Y, Z). Скомпилируйте и запустите программу, и вы должны увидеть свою фигуру... однако это не совсем квадрат?

Делаем квадрат квадратным

Проблема с неквадратным квадратом появилась из-за предположения OpenGL, что область просмотра представляет собой квадрат, а на устройствах с Android экран обычно совсем не квадратный. Чтобы это исправить, нам понадобятся проекционная матрица и матрица обзора камеры, с которыми мы правильно вычислим координаты и привяжем их к экрану вашего устройства. Более подробно о матрицах написано во врезке; проекционная матрица по сути связывает «идеальный квадрат» OpenGL с реальным экраном, а матрица обзора камеры представляет визуализируемые объекты так, как их видит камера с заданного положения. Наша камера находится по центру экрана.

Все эти действия выполняются в `CISGLRenderer.java`. Для начала нужно изменить код вершинного шейдера, добавив туда ссылку на единственную матрицу `uMVPMatrix`.

```

Мы объединили обе матрицы в одну:
private final String vertexShaderCode =
    "uniform mat4 uMVPMatrix; \n" +
    "attribute vec4 vPosition; \n" +
    "void main() { \n" +
    "    gl_Position = uMVPMatrix * vPosition; \n" +
    " } \n";

```

Здесь добавляется новая строка с матрицей, затем она умножается на `vPosition` для вычисления `glPosition`.

Также добавим несколько приватных членов матрицы для хранения разных данных:

```

private int.mvpMatrixHandle; // matrix handle
private float[] uMVPMatrix = new float[16]; // объединенная матрица
private float[] vMatrix = new float[16]; // матрица обзора камеры
private float[] projMatrix = new float[16]; // проекционная матрица

```

Теперь допишем немного кода в `onSurfaceCreated()` и `onSurfaceChanged()`:

```

public void onSurfaceCreated(GL10 gl, EGLConfig config) {
    [...]
   .mvpMatrixHandle = GL ES20.
   .glGetUniformLocation(program, "uMVPMatrix");
    [...]
}

public void onSurfaceChanged(GL10 gl, int width, int height)
{
    GL ES20.glViewport(0, 0, width, height);
    float ratio = (float) width/height;
    Matrix.frustumM(projMatrix, 0, -ratio, ratio, -1, 1, 2, 7);
    Matrix.setLookAtM(vMatrix, 0, 0, 0, -3, 0f, 0f, 0f, 0f, 1.0f, 0.0f);
}

```

Подключаемся к Матрице

Методы класса Matrix позволяют выполнять различные манипуляции с матрицами. `frustumM()` задает нашу проекционную матрицу в виде шести плоскостей отсечения. Плоскость отсечения в трехмерной графике говорит визуализатору, «на каком расстоянии» от наблюдателя он должен прекратить вычисления поверхностей. Это означает, что для самых далеких поверхностей вычисления не выполняются; таким образом, мы экономим вычислительные ресурсы.

Левая и правая плоскости отсечения задаются переменной `ratio`, это левый и правый края экрана. Нижняя и верхняя граница экрана задаются координатами -1 и 1. Два последних значения – «ближняя» и «дальняя» граница, у нас это 2 и 7 (они должны быть положительными). Мы получаем унифицированный дескриптор матрицы для последующего использования, а затем задаем

Скорая помощь

`android.opengl.GLES20` по сути представляет собой интерфейс к библиотекам OpenGL, поэтому документацию лучше искать на сайте www.opengl.org/sdk/docs/, а не в документации по Android.



➤ **Великолепный квадрат! Который... вообще-то не квадрат. В следующем разделе мы это исправим.**

Проекции, камера и матрицы

Как вы, возможно, заметили, все наши матрицы имеют размер 4×4 . Пока что мы пользовались тремя значениями для хранения положения вершины (с координатой z , равной нулю; для трехмерных фигур она будет ненулевой); что же в четвертом? На самом деле, у вершин в OpenGL всегда четыре координаты: (x, y, z, w) . Подробный рассказ об этом вне рамок нашего урока, но координата w по существу позволяет реализовать перемещения (перемещения в пространстве, не только повороты или масштабирование) с помощью матричных операций. Если вы хотите заниматься 3D-графикой,

стоит познакомиться с математическими основами матриц, но они тоже вне рамок нашего урока. На следующем уроке я расскажу о них чуть больше.

Пока вы можете рассматривать матрицы как способ хранения действий, которые можно применить к объекту. Проекционная матрица берет объект и изменяет его в соответствии с проекцией. Матричные методы вроде **frustumM** позволяют задавать значения проекционной матрицы с помощью плоскостей отсечения, а не только работая с матрицей напрямую. Аналогично, матрица обзора камеры задается путем указания положения камеры

и ее восходящего вектора. Матрица хранит эту информацию, дожидаясь, пока OpenGL нарисует объект. Перед прорисовкой объекта его координаты умножаются на матрицу, и получается новый набор координат, определяющих, как объект должен выглядеть в пространстве с заданными камерой и проекцией.

В общем, можно добавить сколько угодно матриц для выполнения всех нужных преобразований над объектом, после чего у OpenGL будут окончательные координаты объекта на экране, и он нарисует его.

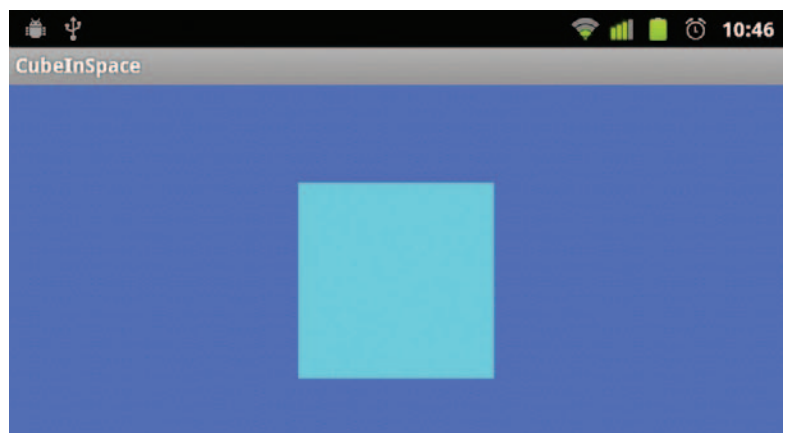
матрицу обзора камеры методом **setLookAtM**. Она будет храниться в **vMatrix** со смещением 0. Точка наблюдения — $(0, 0, -3)$, $x = y = 0$ и $z = -3$, т.е. точка наблюдения выходит «за» экран на три единицы системы координат.

Центр области задан в $(0f, 0f, 0f)$, т.е. совпадает с центром экрана, а восходящий вектор — в $(0f, 1.0f, 0.0f)$. Восходящий вектор задает направление «вверх»; здесь это, как обычно, ось Y .

Наконец, изменим метод **onDrawFrame()** и воспользуемся всеми новыми матрицами:

```
public void onDrawFrame(GL10 gl) {
    GLES20.glClear(GLES20.GL_COLOR_BUFFER_BIT |
        GLES20.GL_DEPTH_BUFFER_BIT);
    Matrix.multiplyMM(uMVPMatrix, 0, projMatrix, 0, vMatrix, 0);
    GLES20.glUniformMatrix4fv(mvpMatrixHandle, 1, false,
        uMVPMatrix, 0);
    GLES20.glUseProgram(program);
    square.draw(positionHandle);
}
```

Метод **multiplyMM** умножает **vMatrix** на **projMatrix** и записывает результат в **uMVPMatrix** (все нули — смещения в результирующем массиве: у нас их нет). Метод **glUniformMatrix4fv** обновляет **mvpMatrixHandle**, который мы получили из вершинного шейдера в методе **onSurfaceChanged()** с помощью **uMVPMatrix**, не транслируя никаких элементов (**false**).



Все готово! Перекомпилируйте программу, и вы должны увидеть настоящий квадрат.

➤ **Теперь квадрат и вправду квадрат.**

Будьте осторожны! Любые ошибки в строках шейдеров OpenGL приведут к тому, что код перестанет работать, но компилятор Android их не увидит, так как для него это просто строки.

Если код работает не так, как ожидалось, внимательно проверьте заглавные буквы, пробелы и корректность используемых операторов в строках.

Добавим цвета

Сейчас мы задаем цвет квадрата вручную, как часть переменной **fragmentShaderCode**. Лучше передавать его извне. Измените переменные **fragmentShaderCode** и **vertexShaderCode**:

```
private final String vertexShaderCode =
    "uniform mat4 uMVPMatrix; \n" +
    "attribute vec4 vPosition; \n" +
    "attribute vec4 aColour; \n" +
    "varying vec4 vColour; \n" +
    "void main(){ \n" +
    "    vColour = aColour; \n" +
    "    gl_Position = uMVPMatrix * vPosition; \n" +
    "} \n";
private final String fragmentShaderCode =
    "precision mediump float; \n" +
    "varying vec4 vColour; \n" +
    "void main(){ \n" +
    "    gl_FragColor = vColour; \n" +
    "} \n";
```

Цвет стал передаваться в **aColour** и **vColour**. Переменные с изменяемыми значениями служат в качестве интерфейса между вершинным и фрагментным шейдерами.

Цвет вершин задается в переменной **aColour**, которая не изменяется. Переменная **vColour** (которая изменяется) в точках вершин равна **aColour**, а затем во фрагментном шейдере интерполируется до цвета пикселей, который меняется между вершинами; так мы получаем разноцветную фигуру.

Затем в методе **onSurfaceCreated()** создадим объект Program, который использует эти значения:

```
private int colourHandle;
public void onSurfaceCreated(GL10 gl, EGLConfig config) {
    [ ... ]
    program = GLES20.glCreateProgram();
    GLES20.glAttachShader(program, vertexShader);
    GLES20.glAttachShader(program, fragmentShader);
    GLES20.glBindAttribLocation(program, 0, "vPosition");
    GLES20.glBindAttribLocation(program, 1, "aColour");
}
```

»

```

GLES20.glLinkProgram(program);
positionHandle = GLES20.glGetAttribLocation(program,
"vPosition");
colourHandle = GLES20.glGetAttribLocation(program,
"aColour");
}

```

Данный код связывает дескриптор положения и дескриптор цвета с соответствующими переменными в вершинном шейдере (**vPosition** и **aColour**) и получает дескрипторы для обеих. Теперь измените вызов **square.draw()** в **onDrawFrame()**, передав в качестве аргументов дескрипторы положения и цвета.

В новой версии **square.draw()** будет всего две новых строки:

```

GLES20.glVertexAttribPointer(colourHandle, 4, GLES20.GL_
FLOAT, false, 0, squareColourBuffer);
GLES20.glEnableVertexAttribArray(colourHandle);

```

Обратите внимание, что мы здесь используем переменную **squareColourBuffer**, которая связана с **colourHandle**.

Понятное дело, именно в ней хранятся цвета вершин:

```

private FloatBuffer squareColourBuffer; float verticesColour[] =
1.0f, 0.0f, 1.0f, 1.0f,
0.0f, 1.0f, 0.0f, 1.0f,
0.0f, 0.0f, 1.0f, 1.0f,
0.0f, 0.0f, 0.0f, 1.0f,
};

```

Для каждой вершины указан цвет в формате RGBA, и пиксели в этой вершине будут в него окрашены. По мере перемещения к следующей вершине визуализатор будет интерполировать цвета между цветами вершин, и один цвет будет плавно перетекать в другой (как на рисунке).

Если все строки (вершины) будут одного цвета в **verticesColour**, мы получим однотонный квадрат.

squareColourBuffer задается в конструкторе точно так же, как и **squareBuffers**. Скомпилируйте и запустите программу, и вы должны увидеть разноцветный квадрат.



➤ Разноцветный квадрат.

Квадрат превращается в куб

Квадраты – это прекрасно, но истинная привлекательность OpenGL – в умении работать с трехмерными фигурами. Создадим класс **Cube**, чтобы нарисовать куб вместо нашего квадрата:

```

public class Cube {
private FloatBuffer cubeBuffer;
private FloatBuffer cubeColourBuffer;
private ShortBuffer cubeIndicesBuffer;
float vertices[] = {
-0.5f, -0.5f, -0.5f,
// Координаты 8 вершин; подробности см. на DVD
};
float verticesColour[] = {
0.0f, 0.0f, 0.0f, 1.0f,

```

```

// Координаты 8 вершин; подробности см. на DVD
};
short[] cubeIndices = {
0, 4, 5,
// Всего 36 индексов в 12 группах по 3 (см. текст); подробности см. на DVD
};
public Cube() {
// Инициализируем cubeBuffer и cubeColourBuffer как делалось для Square.java
cubeIndicesBuffer = ByteBuffer.allocateDirect(
cubeIndices.length * 4).
order(ByteOrder.nativeOrder()).asShortBuffer();

```

Куб в движении

Чтобы закончить нашу историю, заставим куб двигаться. Для этого нам снова достаточно лишь матрицы движения, но на сей раз в методе **onDrawFrame()**:

```

Matrix.rotateM(mMatrix, 0, 1, 6, 2, 3);
Matrix.setIdentityM(uMVPMatrix, 0);

```

Каждый раз при перерисовке кадра (периодичность перерисовки будет зависеть от аппаратной начинки устройства и от того,

что еще происходит в системе) куб будет поворачиваться на один градус вокруг оси (6, 2, 3). Опять же, попробуйте изменить эти числа, чтобы понять, что происходит (например, увеличьте «градусы», чтобы куб завертелся быстрее). Учтите, что порядок множителей при умножении матриц в **onDrawFrame()** имеет значение; попробуйте поменять местами строку **multiplyMM mMatrix** со строками **vMatrix** и **projMatrix**, и вы поймете, что я имею в виду!

```

    cubelIndexBuffer = cbb.asShortBuffer().put(cubelIndices).
    position(0);
}
public void draw(int positionHandle, int colourHandle) {
// Установим позицию и цвет VertexAttribPointers как
в Square.java
    GLES20.glDrawElements(GLES20.GL_TRIANGLE_STRIP,
    cubelIndices.length, GLES20.GL_UNSIGNED_SHORT,
    cubelIndexBuffer);
}
}

```

Главное отличие в том, что здесь используются координаты вершин куба. В массиве **vertices** определяется 8 вершин куба, которые можно пронумеровать от 0 до 7.

В массиве **cubelIndices** они группируются по три; каждая группа задает треугольник размером в половину грани куба. Это делается потому, что для рисования куба мы воспользуемся примитивом **GL_TRIANGLE_STRIP** (к сожалению, Android не поддерживает **GL_QUADS**, с которым мы могли бы определить грани куба, сгруппировав вершины по четыре).

Такое структурирование данных о вершинах/гранях и метод **glDrawElements()** позволяют задать куб с меньшим количеством вызовов.

Вызвав **cube.draw()** в **CISGLRenderer.onDrawFrame()**, вы все еще увидите квадрат – плоскую переднюю грань куба, на которую вы смотрите спереди. Чтобы увидеть трехмерность куба, его нужно немного повернуть.

Для этого воспользуемся еще одной матрицей **mMatrix** и добавим несколько строк в **onSurfaceCreated()** и одну строку в **onDrawFrame()**:

```

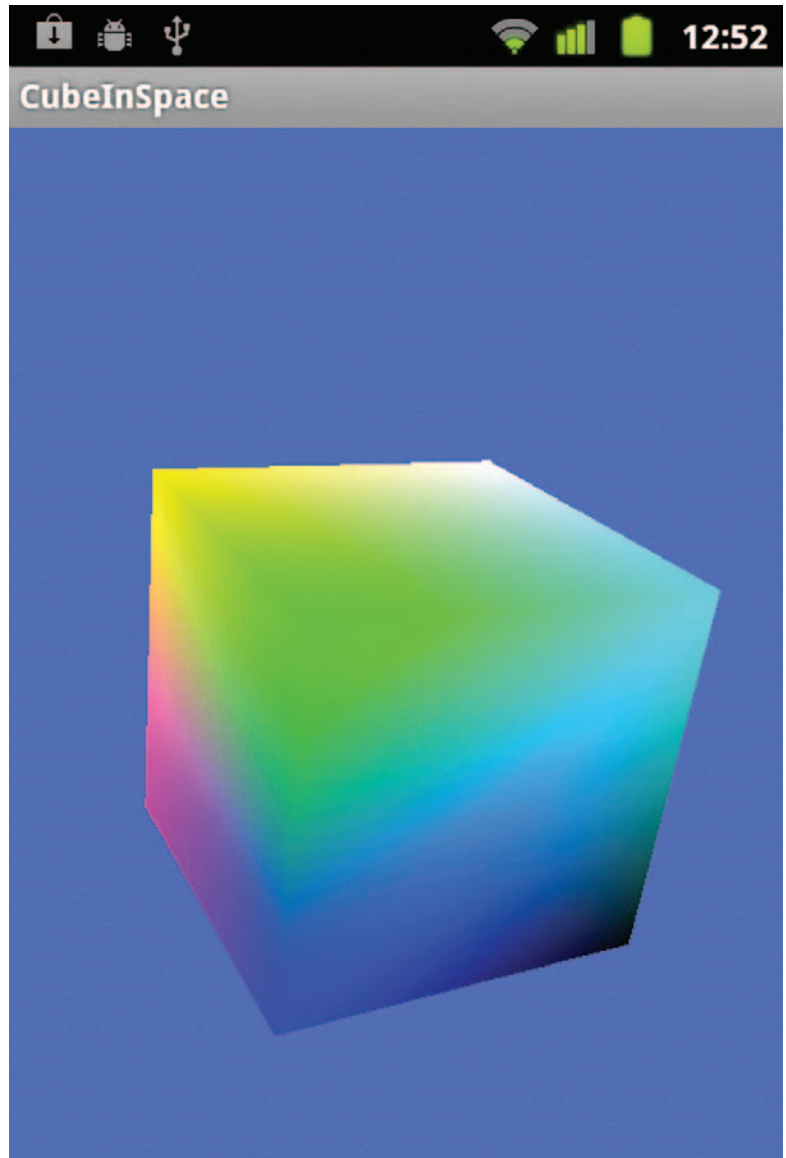
private float mMatrix = new float[16];
public void onSurfaceCreated(GL10 gl, EGLConfig config) {
[ ... rest of method as stands ... ]
    Matrix.setIdentityM(mMatrix, 0);
    Matrix.rotateM(mMatrix, 0, -40, 1, -1, 0);
}
public void onDrawFrame(GL10 gl) {
    GLES20.glClear(GLES20.GL_COLOR_BUFFER_BIT | GLES20.
    GL_DEPTH_BUFFER_BIT);
    Matrix.setIdentityM(uMVPMatrix, 0);
    Matrix.multiplyMM(uMVPMatrix, 0, mMatrix, 0, uMVPMatrix, 0);
    Matrix.multiplyMM(uMVPMatrix, 0, vMatrix, 0, uMVPMatrix, 0);
    Matrix.multiplyMM(uMVPMatrix, 0, projMatrix, 0, uMVPMatrix, 0);
    GLES20.glUniformMatrix4fv(mvpMatrixHandle, 1, false,
    uMVPMatrix, 0);
    GLES20.glUseProgram(program);
    cube.draw(positionHandle, colourHandle);
}

```

В методе **onSurfaceCreated()** мы инициализируем эту матрицу как единичную, затем поворачиваем ее на месте на -40 градусов вокруг оси XYZ (1, -1, 0) (попробуйте изменить эти числа и посмотрите, что происходит, если вы не знакомы с трехмерным поворотом).

Теперь в методе **onDrawFrame()** нужно умножить эту матрицу на матрицу обзора камеры (**vMatrix**) и на проекционную матрицу (**projMatrix**). Это означает, что нужно выделить строки и умножить каждую из них по очереди для получения объединенной матрицы, но по сути мы делаем здесь то же, что и раньше (объединяем различные матрицы в одну, чтобы применить ее к нашей фигуре), только с еще одной матрицей. Потом мы рисуем наш куб.

У нас должно получиться нечто похожее на куб, но его грани выглядят немного странно. Чтобы они смотрелись более солидно, добавьте еще две строки в метод **onSurfaceCreated()** после вызова **glClearColor()**:



▶ Разноцветный вращающийся куб.

```

GLES20.glEnable(GLES20.GL_DEPTH_TEST);
GLES20.glDepthFunc(GLES20.GL_LEQUAL);

```

В первой строке включаются проверка глубины и буфер глубины. С включенным буфером глубины каждый раз при прорисовке пикселя его значение глубины сравнивается с сохраненным значением глубины.

Мы пользуемся параметром **GL_LEQUAL**, поэтому новый пиксель будет нарисован только в том случае, если он ближе к наблюдателю, чем старый, и, таким образом, виден ему.

Без этого, так как OpenGL рисует пиксели в любом порядке, можно получить причудливые результаты и частично видеть объекты «насквозь». Скомпилируйте и запустите программу и полюбуйтесь своим кубом! **LXF**

В следующий раз...

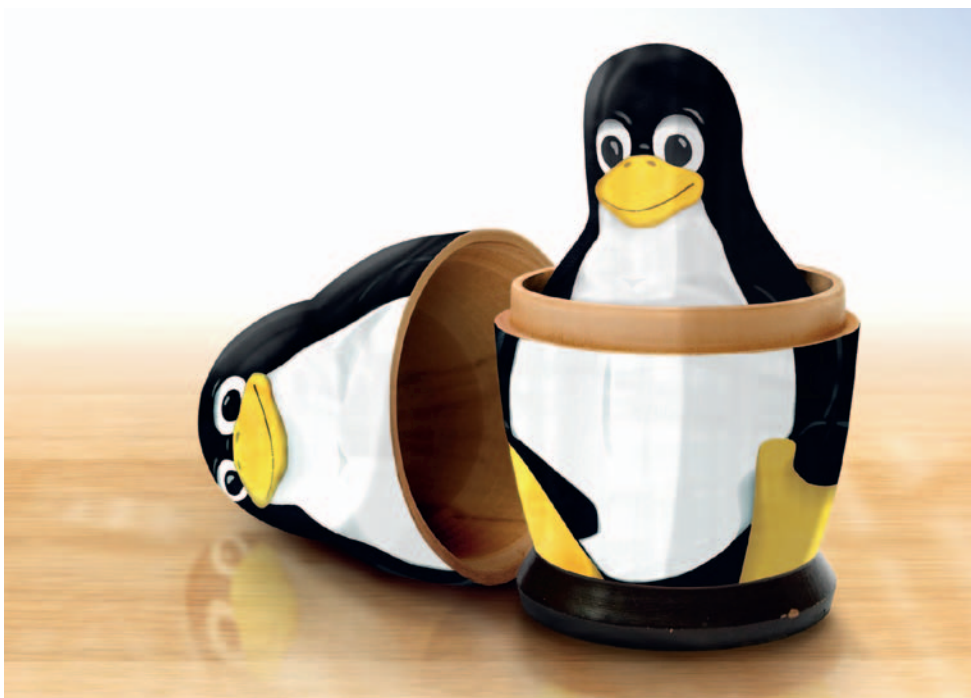
На следующем уроке мы будем перемещать куб по экрану в ответ на касания сенсорного экрана и познакомимся с основами добавления текстур на двумерные и трехмерные объекты.

Сравнение

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

Приложения для виртуализации

Маянк Шарма совлек покров тайны с виртуализации, сравнивая приложения, предназначенные для обычных пользователей настольных ПК.



Про наш тест...

Мы взяли самые свежие релизы программ с их официальных сайтов. Для тестирования проприетарных *VMware Workstation* и *Parallels Workstation* использовалась бесплатная 30-дневная версия. Все программы были установлены на Fedora 17 и Ubuntu 12.04.

Кроме *Virtual Machine Manager (VMM)*, все программы тестировались на одноядерном и многоядерном ПК. *VMM* — это просто интерфейс для управления виртуальными машинами (ВМ), созданными гипервизором, например, *Kernel Virtual Machine*, которая требует такой виртуализации оборудования, как Intel VT-x. На ВМ пробовались Windows 8, Windows XP, Fedora 17 с Gnome и KDE, Ubuntu 12.04 с Unity и AmigaOS-совместимый Icaros Desktop.

На поддерживаемые гостевые системы мы также установили гостевые дополнения.

Наша подборка

- » Parallels Workstation
- » VirtualBox
- » Virtual Machine Manager
- » VMware Player
- » VMware Workstation

Возможно, вы этого не осознаете, но железо вашего настольного ПК, вероятно, мощнее, чем первые суперкомпьютеры. А может, и нет. Но в любом случае, мы хотели бы сказать, что многоядерный компьютер на вашем столе способен на гораздо большее, чем вы от него получаете.

Один из наилучших способов использовать огромное избыточное количество ОЗУ и ядра CPU — это использовать их для создания и работы виртуального оборудования. Виртуализация уже перестала быть темным магическим искусством чародеев, таящихся в серверных, каким она была раньше. Она проложила себе дорогу

на ПК с Linux благодаря программам наподобие *VirtualBox*.

Виртуализация сильно упрощает не только нашу жизнь в Башнях LXF — она поможет и вам. Начинающие могут воспользоваться ею для тестирования дистрибутивов, которые мы ежемесячно выкладываем на LXF DVD. Все программы виртуализации могут загружаться из ISO-образов, и их даже не надо записывать

на оптический носитель. Но виртуализация умеет намного больше, чем просто создавать изолированную среду для тестирования программ. Самые последние программы виртуализации имеют фантастические 3D-возможности и могут выполнять сложные с графической точки зрения задачи. И если вам нужен второй компьютер, рекомендуем совершить тест-драйв некоторых из этих программ.

«Виртуализация упрощает не только нашу жизнь в Башнях LXF — она поможет и вам.»

Полезные функции

Физические возможности виртуального ПО.

Важнейшими характеристиками программы виртуализации являются список хостовых ОС, на которых она может работать, и гостевых ОС, которые она может запускать на своем виртуальном оборудовании. В этом смысле особого выбора среди программ нет, поскольку почти все они поддерживают основные ОС как в качестве хостовых, так и в качестве гостевых.

Хотя *Parallels Workstation 6* являет собой странное исключение. Официально она поддерживает меньше всего операционных систем. *VMware Workstation 9*, *Player 5* и *VirtualBox 4.2* добавили поддержку для Windows 8, а *Parallels Workstation* не сделала этого. Фактически, Fedora 15 и Ubuntu 11.04 – самые последние гостевые ОС, которые она поддерживает.

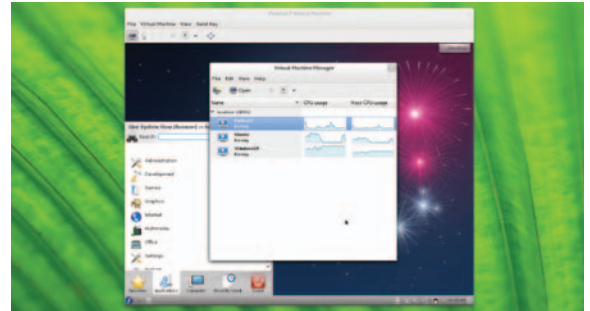
Но если вы готовы смириться со столь ограниченным выбором ОС, то *Parallels* предложит вам несколько весьма интересных функций. Вы можете создавать зашифрованные VM и VM, которые следуют предварительно заданным ограничениям по ресурсам, чтобы не перегружать хост.

Способность *Parallels* делать автоматические запрограммированные на опре-

деленное время моментальные снимки тоже весьма полезна и проста в настройке. Нужно только указать периодичность создания моментальных снимков и сколько снимков надо сохранять в своей библиотеке. На основе этих двух параметров *Parallels* предлагает схему восстановления. То есть если вы делаете моментальные снимки системы ежедневно и разрешаете сохранить 10 снимков, приложение позволит восстановить 4 ежедневных снимка, 3 еженедельных и 3 ежемесячных. Программа также обладает полезными опциями безопасности. Она может запускать VM в режиме Safe, возвращая гостевую ОС к ее исходному состоянию в момент перезагрузки, а также изолировать VM от хоста, чтобы они не делили доступ к папкам и приложениям.

Ограниченная VM от *VMware Workstation* не дает пользователям менять свои настройки и блокирует доступ к определенным ресурсам, таким, как устройства USB. Более того, *VMware Player* может запускать машины с ограниченными функциями, созданные *VMware Workstation*.

VMware Player делает многое из подвластного *VMware Workstation*. Однако здесь



нет того, что в терминологии *VMware* обозначено как «предназначенные специально для разработчика функции» – например, многочисленных моментальных снимков системы и клонов. В этой области *Player* уступает *VirtualBox*, где имеются полные реализации некоторых функций, в *VMware Player* присутствующие в урезанном виде. *VirtualBox* может создавать моментальные снимки и клоны VM, либо посредством создания полной копии VM, либо используя моментальный снимок в качестве базы для клона.

VirtualBox также превосходит *VMware Workstation*, поскольку предлагает вам указать границы использования хостового CPU. То есть вы можете изменить настройки VM в *VirtualBox*, чтобы использовать большую часть ресурсов вашего процессора, и разрешить ему одновременно использовать несколько процессоров или ядер процессора.

» В *Virtual Machine Manager* можно посмотреть статистику производительности и потребления ресурсов для каждой VM и хоста.

Вердикт

VMware Workstation	★★★★★
VirtualBox	★★★★★
Parallels Workstation	★★★★★
VMware Player	★★★★★
Virtual Machine Manager	★★★★★

» VMware Workstation имеет даже больше функций, чем HADO.

Поддержка оборудования

...реального и виртуального.

Помимо оборудования, которое можно виртуализовать, другим важным критерием при выборе программы виртуализации является то, насколько полно и хорошо она использует ваше родное оборудование.



» *Parallels* и *VMware Workstation* могут создавать VM, которые ограничивают доступ к определенному оборудованию.

Все программы в нашем Сравнении умеют создавать VM с несколькими виртуальными CPU и множеством жестких дисков и ОЗУ, предполагая, что у вас хватит физических ресурсов для создания их резервных копий. Также эти программы виртуализуют широкий ряд устройств, имеющихся во многих современных системах ПК, в том числе контроллеры жестких дисков IDE, SCSI и SATA, несколько виртуальных сетевых карт и звуковых карт, виртуальные серийные и параллельные порты и Input/Output Advanced Programmable Interrupt Controller (I/O APIC).

Более того, *VMware Workstation 9* имеет графический драйвер монитора, способный делать рендеринг 3D в Windows 8 без ускорителя оборудования. По сути, *VMware* разработали графический драйвер OpenGL и отправили его в *X.Org*, что дало пользователям доступ к усовершенствованному графическому возможностям *VMware* без установки *VMware Tools*.

VirtualBox поддерживает ускорение 3D-графики, но только после установки проприетарных сторонних дополнений. VM *VirtualBox* также поддерживают разрешения экрана, многократно превосходящие разрешение физического экрана, что позволяет распределить их по множеству экранов, присоединенных к системе-хосту.

Parallels Workstation также поддерживает несколько дисплеев, и вы можете задать порядок управления сигналами мыши с помощью менеджера дисплеев. Можно также направлять информацию на печать с VM на принтеры с распределенным доступом, сетевые принтеры, принтеры USB и даже порты LPT.

Однако поддержка USB 3.0 есть только в *VMware Workstation* и *Player*. *VMware Player* обладает всеми усовершенствованиями виртуального оборудования, включенными в *Workstation*, и, в отличие от предыдущих версий, предлагает настройку сетевых адаптеров для каждой VM.

Вердикт

VMware Workstation	★★★★★
VMware Player	★★★★★
Parallels Workstation	★★★★★
VirtualBox	★★★★★
Virtual Machine Manager	★★★★★

» И снова парни из VMware превзошли остальных игроков на поле.

Удобство использования

Нужна ли докторская степень, чтобы пользоваться этими программами?

Виртуализация – это устрашающий термин, который издавна вгонял пользователей в трепет. Именно поэтому, хотя настройка виртуальной машины в наши дни сильно отличается от того, чем она была раньше, в силу того, что мы оцениваем программы, предназначенные для пользователей настольных ПК, мы не рекомендовали бы слишком сложные в освоении.

Сама по себе установка программ не является проблемой. Если программы нет в репозитории вашего любимого дистрибу-

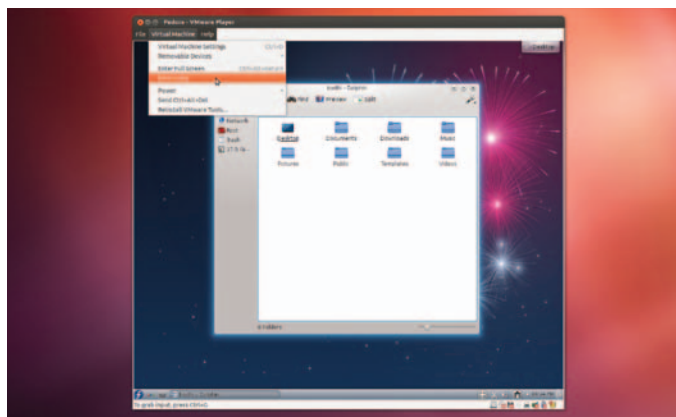
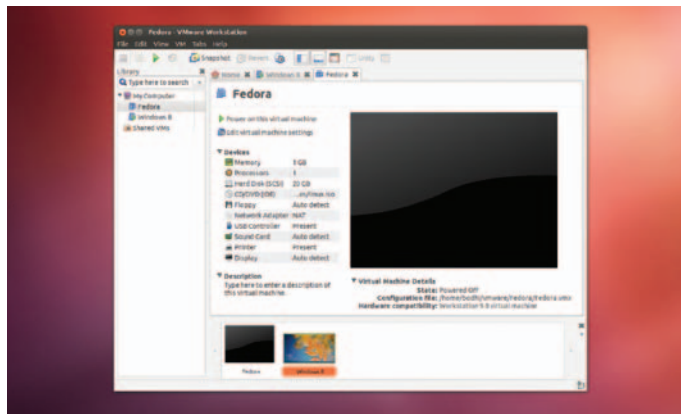
тива, можете скачать ее с ее сайта в виде простого для установки пакета. Самая важная и основная часть работы с программами виртуализации – создание и управление виртуальной машиной. И хотя каждая программа в нашем списке снабжена мастером создания VM, мы проследим за процессом добавления и настройки виртуального оборудования и за производительностью гостевой ОС до и после установки соответствующих дополнительных пакетов.

VMware Workstation ★★★★★

Программа предоставляет вам два способа создания VM. Рекомендуемый способ – пятишаговый процесс, который спрашивает вас, хотите ли вы установить ОС с физического диска или в качестве ISO-образа, тип и версию ОС, а также имя и расположение VM.

Единственный вопрос о виртуальном оборудовании, который будет вам задан, касается размера виртуального диска и того, хотите ли вы поделить его на несколько файлов или оставить в виде одного файла. Да, в конце вы получаете возможность индивидуальной настройки оборудования, но если вы действительно хотите это сделать, выбирайте лучше второй способ в начале процесса.

Workstation установила Windows 8, ни разу не споткнувшись, вместе с *VMware Tools*. Однако установка инструментов на Linux довольно неудобна.



VMware Player ★★★★★

Как мы уже отметили ранее, *VMware Player* следует рассматривать как урезанную версию *Workstation*. Хотя ее интерфейс отличается от *Workstation*, процесс создания виртуальной машины в точности тот же. Единственное отличие – можно настроить виртуальное оборудование отдельно от виртуального жесткого диска в конце процедуры. *Player* может виртуализировать то же оборудование, что и *Workstation*, и дает вам те же возможности по индивидуальной настройке оборудования. В плане производительности между этими двумя тоже не наблюдается особой разницы.

Хотя в *Player* отсутствует возможность делать моментальные снимки, мы смогли скопировать физическую директорию, где она хранилась на хосте, и позднее восстановить ее, повозившись с изначальной VM. Это довольно дубовый способ сохранения моментальных снимков здоровой VM, но он работает.

Документация и поддержка

Отзовутся ли они на ваш сигнал SOS?

Подобно всем коммерческим программам, *VMware Workstation* и *Parallels Workstation* имеют специальную команду поддержки, поддерживаемую обширной инфраструктурой.

VMware выдает подробную документацию Getting Started для обоих своих решений; для *Workstation* также имеется подробное руководство пользователя. Есть документация по установке гостевых ОС и руководство по установке на гостевые системы дополнений *VMware Tools*.

VMware имеет сайт для сообщества поддержки своих продуктов, в том числе *Workstation* и *Player*. И если у *Player* имеется только форум, то сообщество *Workstation* предлагает ссылки на большое количество документов и FAQ. Пользователи *Workstation* также могут зарегистрироваться для прохождения курса онлайн-обучения.

Parallels Workstation предлагает массу справочной документации, в том числе Getting Started и подробное руководство пользователя. Кроме того, есть многочис-

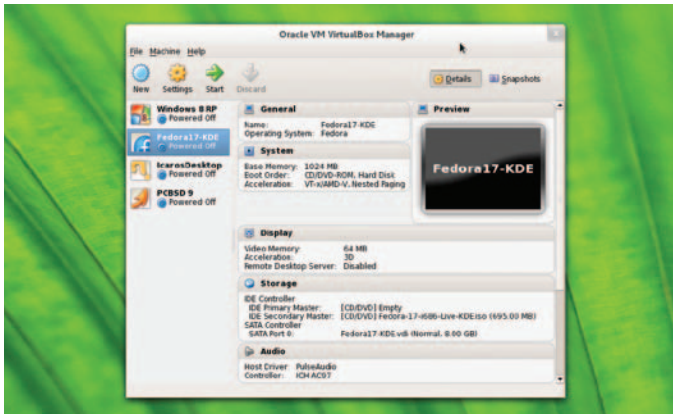
ленные возможности поддержки, если вам нужна рука помощи. В *VirtualBox* имеется всесторонняя документация для конечного пользователя и инфраструктура поддержки с активными форумами, списками рассылки и каналом IRC. И если вы готовы платить, Oracle также предлагает коммерческую поддержку для *VirtualBox*.

А вот у поддерживаемой RedHat *Virtual Machine Manager* есть только Wiki с ограниченным числом статей, список рассылки и сайт <http://virt-tools.org/>.

Вердикт

Parallels Workstation	★★★★★
VMware Workstation	★★★★★
VirtualBox	★★★★★
VMware Player	★★★★★
Virtual Machine Manager	★★★☆☆

» VirtualBox предлагает много информации из серии «Сделай сам».



VirtualBox ★★★★★

Создание VM в *VirtualBox* не особо отличается. Вы начинаете с выбора имени VM, а также типа и версии ОС, которую вы будете устанавливать на нее. Затем настраиваете очередь виртуальной памяти и жесткого диска.

У виртуального жесткого диска имеется собственный мастер настройки, и он начинается с того, что предлагает вам выбрать формат виртуального диска. *VirtualBox* поддерживает форматы дисков всех программ виртуализации, в том числе *VMWare* и *Parallels*.

Если вам надо настроить другое оборудование, придется делать это после создания VM. Интересно, что при редактировании VM окно Settings предупредит вас, если определит неоптимальную настройку: скажем, если вы ответили более половины своего физического ОЗУ под VM. Это очень удобно, особенно для новых пользователей.

Parallels Workstation ★★★★★

Перед установкой *Parallels Workstation* предложит вам либо отключить SELinux, либо включить его в режим Permissive. Программа не устанавливается на самую последнюю версию Fedora 17, но безупречно работает на неподдерживаемом Ubuntu 12.04. Процедура создания VM более или менее такая же, как у остальных, за исключением опции Express Install для гостевых Windows. Также, хотя она и устанавливает Windows 8 Release Preview без всяких жалоб, у VM возникают проблемы с графикой при работе в полноэкранном режиме, даже с установленными инструментами.

Производительность гостевых систем Linux без инструментов на удивление плоха, с жутко крупным разрешением и неровными движениями мыши. Установка инструментов на некоторые гостевые системы Linux, например, на Fedora 15 KDE, требуют наличия дополнительных компонентов, и спустя примерно полчаса внезапно обрываются сообщением об ошибке.

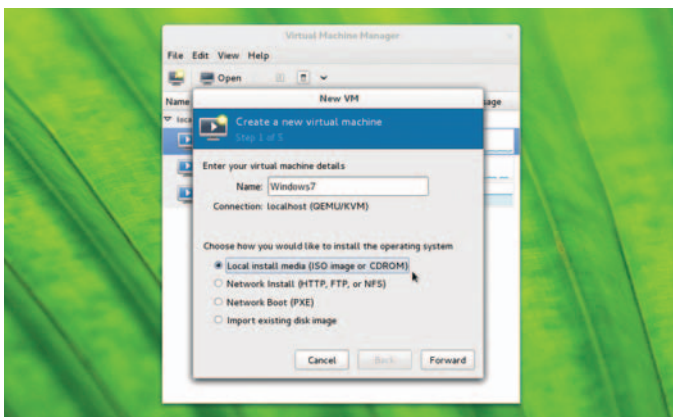


Virtual Machine Manager ★★★★★

Эта программа дает вам намного больше опций при создании VM. Например, вы начинаете с соединения с гипервизором – в нашем случае это *KVM*. Помимо обычных средств установки, таких, как ISO-образ, можно также сделать сетевую установку через HTTPS, FTP и NFS.

В отличие от остальных программ, она по умолчанию сразу же отводит весь диск целиком под VM. Чтобы настроить виртуальное оборудование, помимо памяти, CPU и сети, вам придется подождать окончания процесса создания VM.

В наших тестах программа отобразила необычайно большое разрешение VM Ubuntu, и не пожелала изменять размер на меньший. После того, как мы поставили VM на паузу, она отказалась возобновлять работу. Даже выключение не помогло: после перезагрузки она немедленно подвесила компьютер. Однако Fedora 17 KDE вполне можно было пользоваться.



Стоимость и лицензии

Страсть к дензнакам?

V *VMware Workstation* не только выглядит впечатляюще, она еще и самая дорогая программа в этом Сравнении. Одна копия *Workstation* обойдется вам в \$249 (около £154). Но если вы уже являетесь пользователем, обновить свою программу до версии *Workstation 9* вы сможете за \$119 (около £74).

Parallels Workstation 6 стоит дешевле – £64,99, но в ней меньше функций по сравнению с решением от *VMware*. И *VMware*, и *Parallels* дают возможность оценить

их продукты бесплатно в течение 30 дней. Но если вам ни к чему некоторые функции и вы можете обойтись без моментальных снимков своих VM или не собираетесь их клонировать, воспользуйтесь *VMware Player 5.0*, доступной бесплатно для личного некоммерческого использования. Копия *VMware Player* также идет с *VMware Workstation*, и может применяться для запуска неограниченных VM, созданных в *Workstation*. С другой стороны, в *VirtualBox 4.2* есть более удобные функции для сред-

него пользователя настольного ПК, чем в *VMware Player*, и основной пакет имеет преимущество – код его открыт. Но чтобы добиться от *VirtualBox* наилучшего результата, придется установить отдельный пакет расширений с компонентами с закрытым кодом, например, поддержку USB 2.0.

И, наконец, *Virtual Machine Manager*, распространяемый под GNU GPL. Он не стоит ни гроша, но ему далеко до функциональности *VirtualBox* или любого из проприетарных решений.

Вердикт

Virtual Machine Manager ★★★★★

VirtualBox ★★★★★

VMware Player ★★★★★

Parallels Workstation ★★★★★

VMware Workstation ★★★★★

» Свободная как слово или бесплатная как пиво? Решать вам.

Альтернативные интерфейсы

...и кое-что для опытных пользователей.

То, что эти программы виртуализации предназначены в первую очередь для пользователей настольных ПК, вовсе не означает, что им нечего предложить опытным пользователям. Хотите ли вы запустить VM на удаленной машине или управлять ею из командной строки, у каждой программы из нашего Сравнения найдутся на это средства.

Помимо обычного интерфейса *VMware Workstation*, вы можете также получить доступ к виртуальным машинам удаленно через VNC. Начиная с *Workstation 9*, программа также предлагает экспериментальный web-интерфейс под названием *WSX*. С помощью *WSX* можно обращаться к VM с общим доступом из лю-

бого браузера, который поддерживает HTML5. Вы даже сможете включать и выключать их из браузера. Поскольку эта функция не требует дополнительных модулей расширения браузера, таких, как Flash, вы можете использовать ее для управления своими VM даже с мобильных телефонов и планшетов. Однако в данный момент эта функция находится на активной стадии разработки и не поддерживается производственными средами.

WSX нет на *VMware Player* по техническим причинам, но *Player* можете также загрузить VMы через VNC.

Еще одна уникальная функция *VMware Workstation* – ее способность записывать аудио- и видеоактивности экрана внутри VM прямо в видеофайл *.avi*, пригодный для презентаций, пошаговых руководств или демонстраций.

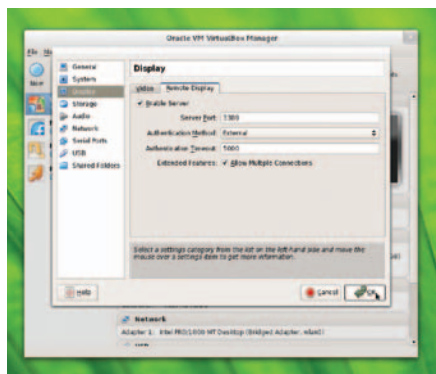
Если вы хотите управлять виртуальными машинами *VMware* из командной строки, можете сделать это через команду *vmrun*, которая также может взаимодействовать с гостевой ОС, например, запускать исполняемые программы в гостевой ОС или делать экранные снимки. CLI *vmrun* работает и для *Workstation*, и для *Player*. Подобным же образом, в *Parallels Workstation* есть утилиты командной строки *prlsrvctl* и *prlctl*, которые применяются для управления раз-

личными конфигурациями *Parallels Workstation* и для выполнения административных задач на VM, например, для установки *Parallels Tools* внутри VM. Также в *Parallels* имеется бесплатное мобильное приложение для iPhone и iPad, которое позволяет запустить, выключить и перезагрузить VM дистанционно.

У *VirtualBox* тоже имеется CLI под названием *VBoxManage*, для управления всеми аспектами программы. Его можно использовать вместе с другими интерфейсами *VirtualBox*, такими, как *VBoxSDL* – это простой графический интерфейс, предназначенный для отладки *VirtualBox*, или *VBoxHeadless* для управления виртуальными машинами, размещенными на сервере без мыши и клавиатуры.

VirtualBox также может отображать VM дистанционно, используя свой закрытый протокол VRDP, совместимый с протоколом Microsoft RDP. Интересно, что *VirtualBox* также поддерживает удаленные устройства USB. И если устройства USB подключены к клиенту, удаленный сервер *VirtualBox* можете получить к ним доступ.

Virtual Machine Manager также устанавливает набор инструментов CLI, таких, как *virt-install* и *virt-clone*, которые можно использовать для управления всеми аспектами VM. С помощью *virt-install* вы можете установить гостевую ОС на VM, а *virt-clone* умеет клонировать VM и копировать образы диска.



➤ Если не работает ни один из интерфейсов, воспользуйтесь *VirtualBox SDK* и напишите собственный!

Вердикт

- VMware Workstation ★★★★★
- Virtual Machine Manager ★★★★★
- VirtualBox ★★★★★
- VMware Player ★★★★★
- Parallels Workstation ★★★★★

➤ У *VirtualBox* должно быть число инструментов для опытных пользователей.

Интеграция

Ладят ли они с хостовой ОС?

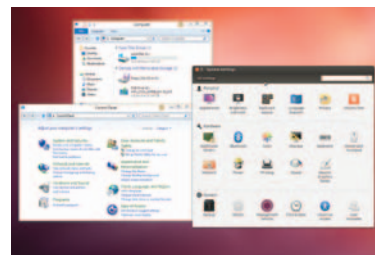
Почти все программы виртуализации сработают с любой ОС, которую вы им предложите. Но работая с ОС из их списка поддержки, вы в полной мере реализуете потенциал своего физического оборудования. Чтобы помочь вам в этом, все пакеты программ виртуализации снабжены специальными инструментами и рядом расширений, которые надо установить на гостевую ОС.

Установив инструменты на гостевую ОС, вы получите расширенную поддержку видео и сможете без помех перемещать мышью с гостевой ОС на хостовую. По установке дополнений вы сможете копировать и вставлять текст и перетаскивать файлы,

перемещая их между гостевой ОС и ОС-хостом. А можно привязать виртуальный диск к имени диска на хосте и вручную копировать файлы с диска или на него.

VMware Workstation и *VMware Player* также позволяют запускать гостевую ОС в режиме Unity, где можно переместить Windows из гостевой ОС в хост-ОС. Но, к сожалению, режим Unity не поддерживается в гостевых системах Linux.

Аналогичная функция под названием Seamless Windows имеется в *VirtualBox*. Она маскирует фон гостевой ОС и отображает Windows в хосте. Особо приятно то, что в равной мере она работает и для гостевых систем Linux.



➤ Функция *VirtualBox Seamless Windows* не так элегантна, как функция в *VMware*.

Реализация *Parallels Workstation* под названием Coherence mode – самая ограниченная, так как работает только на хостах Windows. Но это неудобство компенсируется большим количеством других инструментов, таких, как *Parallels Transporter*.

С помощью этого инструмента вы можете быстро превратить физическую машину, даже удаленную, в виртуальную всего за пару щелчков мыши.

Вердикт

- Parallels Workstation ★★★★★
- VirtualBox ★★★★★
- VMware Player ★★★★★
- VMware Workstation ★★★★★
- Virtual Machine Manager ★★★★★

➤ Выбор среди опций в общем невелик.

Приложения для виртуализации Вердикт

Сообщество Linux смотрит на Oracle косо, но его приложение *VirtualBox* превосходит любое другое приложение виртуализации для десктопа.

Конечно, у *VMware Workstation* множество дополнительных функций, но она слишком дорога по сравнению с бесплатной *VirtualBox*, где есть все основные функции, нужные среднему настольному пользователю. По сути, их достаточно, чтобы превзойти урезанный *VMware Player*.

VirtualBox имеет преимущество в плане стоимости и перед *Parallels Workstation*, однако проприетарное приложение не стоит нашей рекомендации из-за его отношения к хостовым системам Linux. Самые интересные и полезные функции в *Parallels Workstation* ограничены только хостами Windows, а список поддерживаемых гостевых ОС изрядно устарел.

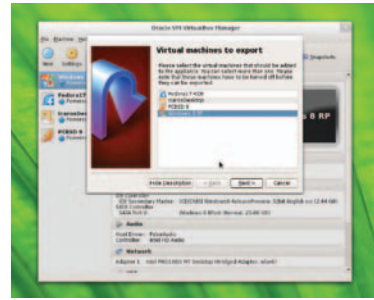
Код ядра приложения *VirtualBox* открыт, но чтобы сделать его в полной мере работоспособным, нужны проприетарные рас-

ширения. А вот *Virtual Machine Manager*, управляющий виртуальными машинами на основе *KVM*, открыт полностью, но зато не столь удобен, как *VirtualBox*.

Есть еще и тот факт, что *VirtualBox* не нужно оборудование с активированной виртуализацией, в отличие от *KVM*. Но она может воспользоваться расширениями оборудования, чтобы создать другой набор виртуального оборудования, например, 64-битные гостевые системы.

Все же кое-чему эта программа может и поучиться у своих соперников. Начать стоит с улучшения режима *Seamless*, сделав его более похожим на режим *Unity* в *VMware*. В настоящее время он выглядит не слишком привлекательно на небольших дисплеях и на тех хостах, где панель расположена внизу.

Но все же программа постоянно совершенствуется. Самая последняя версия, *VirtualBox 4.2*, обладает способ-



» *VirtualBox* может также работать на более старых устройствах, без виртуализации оборудования.

ностью группировать связанные VM, что облегчает управление ими. И хотя они, возможно, и приподзились с этой функцией, но, работая в *VirtualBox* на хосте Linux, вы теперь сможете перетаскивать файлы из хостовой ОС на гостевые ОС Linux. Радостно предвкушаем, что эта возможность также будет распространяться и на другие комбинации хостовых и гостевых ОС в последующих релизах.

И в заключение отметим, что именно удобство и гибкость *VirtualBox* и есть те качества, которые удовлетворяют и обычного, и опытного продвинутого пользователя.

«Удобство *VirtualBox* удовлетворит и обычного, и продвинутого пользователя.»

I VirtualBox ★★★★★
 Сайт: www.virtualbox.org
 » Бесплатная, и работает именно так, как обещано.

IV VMware Workstation ★★★★☆
 Сайт: www.vmware.com/products/workstation
 » Масса функций, но дорого. Зачем платить за то, что излишне?

II VMWare Player ★★★★☆
 Сайт: www.vmware.com/products/player
 » Бесплатная; функций меньше, чем в Workstation или VirtualBox.

V Parallels Workstation ★★★☆☆
 Сайт: www.parallels.com/products/workstation
 » Платная, и ограничивает достойные функции хостами Windows.

III VM Manager ★★★★☆
 Сайт: www.virt-manager.org
 » Бесплатная и быстрая, но новичкам покажется примитивной.

Обратная связь
 Вы согласны с нашим выбором? Напишите нам на lxf.letters@futurenet.co.uk.

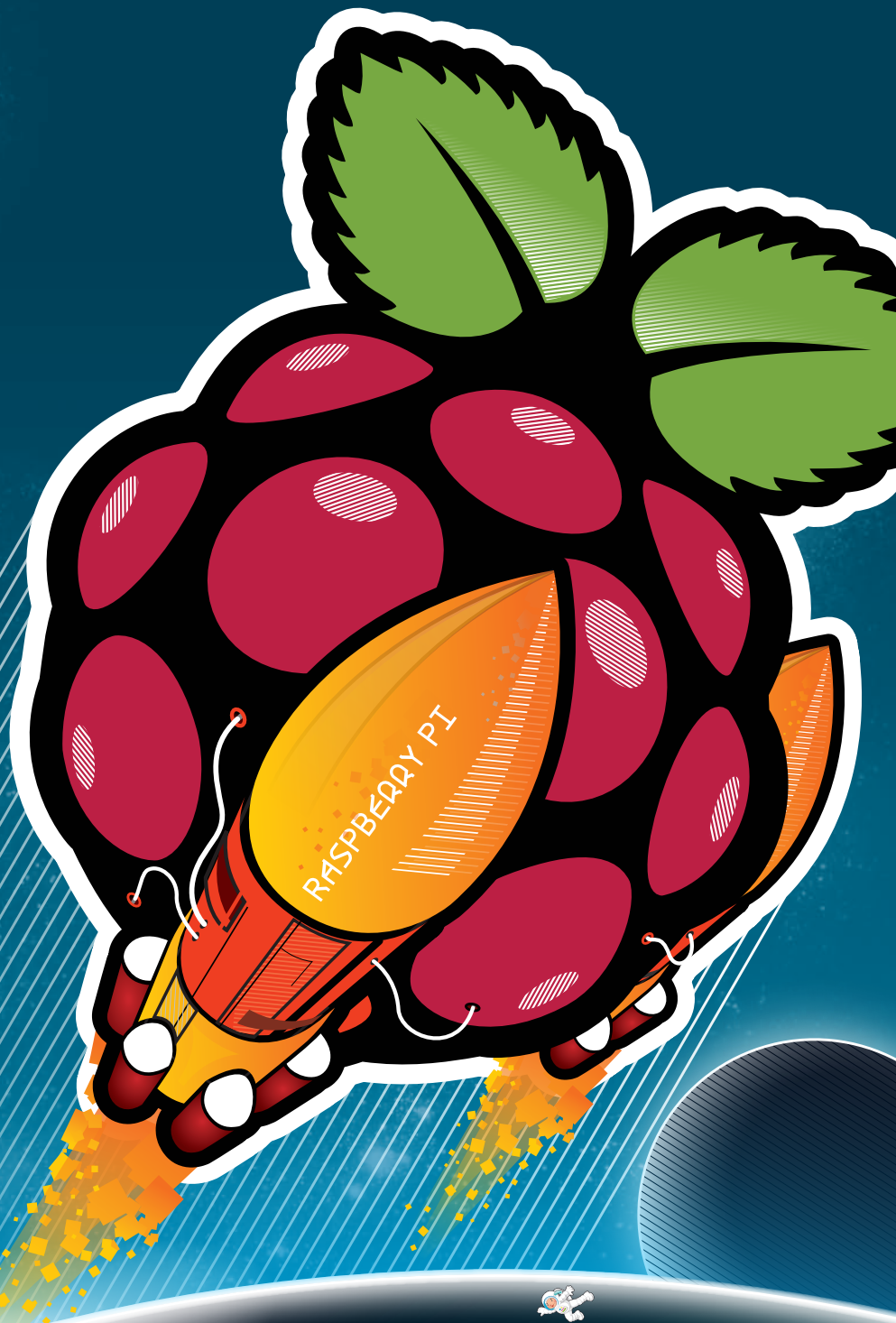
Рассмотрите также...

Поверьте, других настоящих опций виртуализации для пользователей настольных ПК просто нет. По крайней мере, способных работать на хостах Linux. Конечно, есть несколько интерфейсов для *Qemu/KVM*, которые вы найдете в репозиториях своего любимого дистрибутива – таких, как *AQEMU*, *Qemu Launcher*, *Qemulator*, *QtEmu* и прочие. Но те пять опций, о которых мы рассказали в нашем Сравнении – это лучшее, что вы можете найти среди удобных и зрелых программ виртуализации для настольных ПК.

А если вы жаждете приключений и хотите выйти за пределы своей зоны комфорта, можете попробовать гипервизор *Xen*, управляемый из командной строки или с помощью *Virtual Machine Manager*.

Есть также и *Linux Containers (LXC)*, способные виртуализировать одно приложение или всю операционную систему, и *OpenVZ*, похожая на *Jails* во FreeBSD. Руководства к обеим программам вы найдете в предыдущих выпусках журнала. **LXF**

Сверхзадача для Raspberry Pi



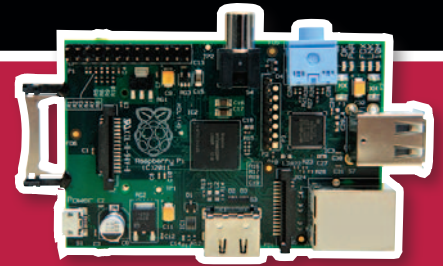
Впервые мы рассказали о Raspberry Pi в феврале, сразу после их выпуска. Мы их сразу полюбили, но не знали, придется ли они по душе всем остальным. Не думаем, что на тот момент кто-то испытывал к ним большую привязанность.

Однако Raspberry Pi задели внутренние струнки любителей со всего мира так, как ни одно устройство за последние годы. Начальная партия количеством 10000 штук разлетелась так быстро, что только самым везучим и убежденным фанатам удалось урвать себе экземпляр, а сейчас они распродают буквально с конвейера. Да чему тут и удивляться – ведь всего за £30 вы получаете рабочий компьютер с Linux, с процессором ARM и 512 МБ ОЗУ (на платах, выпущенных после 15 октября), размером с кредитную карту.

Бен Эверард со своим верным паяльником научит вас извлечь из Pi максимум.

Цель данного проекта – революционизация отсталого компьютерного образования в Великобритании, но пока рано судить, будет ли она достигнута. Одно можно сказать уверенно: миру любительского хакерства оборудования прежним уже никогда не быть. Эти крошечные, но полнофункциональные системы идеальны для внесения возможности обработки данных в необычные места, где в большом дефиците пространство и электроэнергия. Их отправля-

ли в космос и готовят для пересечения океанов, но с тем же успехом они применяются в домашних пивоварнях и для дистанционного управления автомобилями. Мы рассмотрим ряд интересных проектов для Pi и технологии превращения Pi в устройство вашей мечты. Благодаря всесторонности и глубине инструментов Linux, Pi легко превратить во что угодно, от настольного компьютера до мультимедиа-центра или контроллера оборудования.



Дистрибутивы

Как вы резонно предполагаете, для Pi имеется масса дистрибутивов, и создается впечатление, что еженедельно появляются все новые и новые. Здесь мы рассмотрим самые популярные, а также парочку свежих.

Устанавливается дистрибутив не совсем так, как на обычном компьютере. Поскольку все работает с SD-карты, нужно всего лишь записать на эту карту новую операционную систему. Проще всего сделать это утилитой командной строки **dd**. Этот инструмент записывает данные, бит за битом передавая их между устройством и файлом (или, при необходимости, между

двумя файлами или двумя устройствами). Дистрибутивы идут в виде файлов-образов (немного похоже на файлы ISO для CD), которые можно записать на диск после распаковывания с помощью

```
sudo dd if=<image-file> of=<sd-card-device>
bs=4k
sudo sync
```

Вторая строка обеспечивает запись всех данных на карту без отсидки в каком-либо буфере. Скажем, на нашем тестовом компьютере, оборудованном двумя жесткими дисками (**sda** и **sdb**), SD-карта появляется, как **dev/sdc**. Если

вы не уверены в том, каким устройством отображается ваша SD-карта, запустите в терминале **df -h**, и вы увидите список всех устройств. Из него должно быть все ясно.

Чтобы сделать резервную копию настройки вашего Raspberry Pi, можете создать новый файл образа, поменяв флажки **if** (input file) и **of** (output file) в команде **dd**. То есть:

```
sudo dd if=<sd-card-device> of=<new-imagefile>
bs=4k
```

Затем этот образ можно сжать с помощью *gzip* или *bzip*, чтобы он не занимал слишком много места на жестком диске.

Raspbian

Этот дистрибутив рекомендует Raspberry Pi Foundation. Если у вас нет веской причины использовать иной дистрибутив, он, вероятно, будет наилучшим выбором. Основан он на Debian Wheezy, и вы мигом сможете установить что угодно из обширных репозиториях Debian.

Средой рабочего стола по умолчанию является *LXDE*, который славен исключительной легковесностью, но некоторые полагают, что он предлагает лишь самую основу.

Для любителей графических прелестей имеется *Xfce*. В нем есть программа *raspi-config*, дающая, вероятно, самый простой способ настройки вашего Pi.

Raspberry Pi создавался для того, чтобы увлечь детей программированием, и Raspbian соответствует этому намерению. Вы обнаружите на его экране Idle (IDE Python) и Scratch (среду программирования для детей). Берите его с сайта www.raspberrypi.org.

Arch Linux

Если Raspbian пытался защитить пользователей от внутренней настройки операционной системы, то Arch Linux призван помочь пользователям досконально разобраться, как функционирует операционная система.

Изначальный образ, доступный на www.raspberrypi.org, включает только базовую систему, чтобы ваш Pi заработал и соединился с сетью. В него не включены многие программы, которые мо-

гут вам понадобиться для работы с системой, например, графическая среда. Необходимую вам информацию вы найдете на <http://bit.ly/9APmgA>.

Превращение Arch из изначального состояния в рабочую систему потребует некоторых усилий, но в процессе вы узнаете, как взаимодействуют внутренние составляющие дистрибутива Linux.

Конечно, только вы сами можете решить, стоит ли Arch таких трудов.

Raspbmc

Может быть, Raspberry Pi и создавался в качестве образовательного инструмента, но любители очень быстро превратили его в игрушку. Данный дистрибутив намерен превратить ваш Pi в мультимедиа-центр, пригодный для управления ТВ. Он основан на XBMC, который позволяет воспроизводить музыку и видео, имеющиеся у вас в виде файлов, или в виде потока из Интернет. Образ можно скачать с www.raspbmc.com. Более

подробная информация по установке и настройке Raspbmc приводится на следующих страницах. Если у вас настроен движок *MythTV*, можете использовать XBMC Pi для обеспечения внешнего интерфейса.

В зависимости от того, что вы намереваетесь воспроизвести, вам, возможно, придется приобрести пакет кодеков для доступа к защищенным патентами алгоритмам видео и аудио.

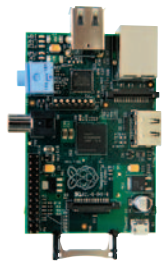
Android

Официальная версия Android Pi находится в разработке (официальное в ней то, что она одобрена Raspberry Pi Foundation). Представлено видео его работы. Когда готовилась эта статья, он не был доступен для скачивания, но на момент вашего чтения ситуация может измениться. Все новости объявляются на www.raspberrypi.org.

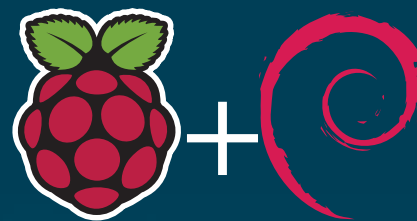
Сообщество работает над неофициальной версией. Произво-

дительность у нее хуже, чем должна быть у официальной (разработчики ее описывают как «просто пригодную к использованию»), зато она доступна уже сейчас.

Если это вас заинтересовало, загляните на сайт проекта – www.razdroid.net.



Raspbian



Для большинства пользователей Raspberry Pi Raspbian является графическим лицом Pi. Его можно найти и установить на SD-карту, следуя инструкциям с предыдущей страницы. Установив и подготовив его к работе, было бы неплохо найти самые свежие версии всех программ: выйдите с вашего Pi в Интернет, откройте терминал и запустите

```
sudo apt-get update
sudo apt-get upgrade
```

Совершенно убойная функция в Raspbian – программа *raspi-config*. Она запустится автоматически при первой загрузке, или ее можно запустить в любое время, введя в терминале **sudo raspi-config**. В ней имеется несколько функций, но важнейшие из них таковы:

» **expand_rootfs** Из-за способа установки Raspbian, тот создает только файловую систему размером 2 Гб, и если ваша карта больше по объему, все оставшееся место будет пустовать. Вы можете использовать эту опцию,

чтобы расширить файловую систему и не дать месту на карте пропасть зря.

» **memory_split** Raspberry Pi использует одинаковый объем памяти для основного процессора и для графического чипа. С помощью данной опции вы можете изменить этот показатель для каждого из них.

» **overscan** Эта опция применяется на некоторых дисплеях, чтобы задействовать для отображения графики все пространство экрана. Ее можно благополучно игнорировать, если не возникнет никаких проблем.

» **overclock** получите дополнительные 50 % производительности без всяких дополнительных затрат! Более подробно вы узнаете об этом из врезки.

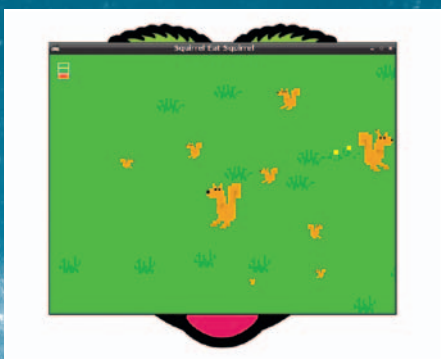
» **boot_behaviour** Эта опция с несколько загадочным названием дает возможность выбо-

ра – загрузить ваш Pi в графической или текстовой среде.

Количество установленных программ минимально. Это хорошая идея, но вы можете обнаружить, что здесь не хватает инструментов, к которым вы привыкли в других дистрибутивах. К счастью, поскольку Raspbian соединен с репозиториями Debian Armhf, у вас есть доступ к большему количеству программ, чем человеку нужно. Мышелюбы, видимо, захотят установить графический менеджер пакетов. Мы рекомендуем *Synaptic*. Чтобы установить его, введите в терминале

```
sudo apt-get install synaptic
```

Затем его можно открыть, перейдя в LXDE menu > Preferences > Synaptic Package Manager. После этого устанавливайте любые программы, какие захотите.



» Как все приличные дистрибутивы, Raspbian предлагает подборку неотразимых игрушек. Это *Squirrels* из подборки *Pygames*.

15 крутых уловок Raspberry Pi

- 1 Превратите Pi в ноутбук: <http://logeeka.ru/raspberry-pi/>
- 2 Используйте в качестве экрана Kindle: www.ponnuki.net/2012/09/kindleberry-pi
- 3 Превратите Pi в суперкомпьютер (почти): <http://bit.ly/OGBGfD>
- 4 Управляйте Arduino через Сеть: <http://bit.ly/Xyjsld>
- 5 Создайте синтезатор: www.raspberrypi.org/archives/1878
- 6 Бродите по ночам: www.aonsquared.co.uk/the_dark_pi_rises
- 7 Управляйте настройкой солнечных батарей: <http://bit.ly/MHwCHF>
- 8 Беседуйте с иностранцем: www.willpowell.co.uk/blog/?p=219
- 9 Отправьте груз в (ближний) космос: www.daveakerman.com/?p=592
- 10 Контролируйте свой дом из Сети: <http://bit.ly/WK1YCR>
- 11 Варите пиво: <http://bit.ly/PSfdzr>
- 12 Учредите школьный класс в коробочке: <http://elinux.org/Rubus>
- 13 Играйте в старые игры: <http://petrockblog.wordpress.com/retropie>
- 14 Учитесь создавать ОС: www.cl.cam.ac.uk/freshers/raspberrypi/tutorials/os
- 15 Создайте будку моментального фото: <http://bit.ly/PvmWfE>

Разгон процессора

Процессор в сердце Raspberry Pi предназначен для работы на частоте 700 МГц, то есть совершения 700 000 000 операций в секунду. Естественно, «предназначен для работы» не означает «обязан работать». Вы можете увеличить эту скорость.

Однако тогда увеличится и энергопотребление, что, в свою очередь, усилит нагрев. Если Pi чересчур перегреется, вы, вероятнее всего, получите вместо рабочего процессора дымящийся кусочек кремния.

К счастью, свежая версия Raspbian (2012-9-18) включает инструмент, который поможет вам нарастить скорость, заодно следя за температурой. Поскольку это официальный инструмент, его применение не нарушит вашей гарантии (в отличие от более ранних неофициальных методов).

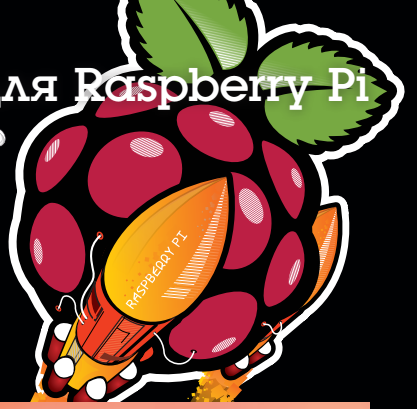
Разгон вашего Pi означает всего-навсего запуск **sudo raspi-config** и выбор Overclocking. Здесь есть несколько опций, из которых предстоит сделать выбор, в зависимости от того, на что вы отважитесь. Если вы заметите, что ваш Pi стал нестабилен, перезагрузитесь, нажав и удерживая клавишу Shift, чтобы отключить разгон, а затем измените настройку, выбрав другую опцию в *raspi-config*. Максимальные параметры дадут вам дополнительные 50 % скорости, что, по нашему мнению, значительно улучшает работу, особенно если речь идет о работе в сети.

Если вам нужно приглядывать за температурой, добавьте виджет Temperature на панель LXDE.

Однако ваш Pi автоматически отключит разгон, если температура достигнет 85 °С.



» Разгон увеличит энергопотребление вашего Pi, способное привести к его нестабильности при работе с несколькими устройствами USB.



RaspBMC

Вы можете установить на Raspbian медиа-плеер, такой как *VLC*, и использовать его для просмотра видео. Это отлично работает, если вы используете свой Pi как обычный компьютер и только периодически наделяете его мультимедийными функциями. Однако малый размер и бесшумность работы устройства делают его отличным выбором для создания собственного развлекательного центра.

Можно начать с Raspbian и настроить его в соответствии со своими потребностями, и это – неплохая мысль, если вы подумываете о каких-то необычных функциях. Однако мы здесь, в *Linux Format*, довольно ленивы и предпочитаем стянуть плоды тяжких трудов других людей, вместо того, чтобы надрываться самим. К счастью для нас (и для вас тоже), команда хакеров уже выполнила самую трудную часть работы по созданию мультимедиа-центра для Raspberry Pi и подготовила пакеты.

Возьмите программу установки с <http://bit.ly/PSFGle> или с диска и скопируйте ее на свой Pi (вам понадобится рабочий Raspbian, откуда мы и начнем). Затем надо просто запустить

```
sudo python install.py
```

чтобы скачать и настроить дистрибутив. Данные на вашей SD-карте будут уничтожены, так что сначала сделайте резервные копии всех важных файлов на другом носителе.

По окончании работы инсталлятора перезагрузите систему. Как только запустится графическая среда, вы увидите разницу.

Здесь используется популярный мультимедийный рабочий стол XBMC, весьма непохожий на *LXDE*, с которым вы, возможно, работали в Raspbian. Вы можете воспроизводить медиа, хранящиеся локально, или, через дополнения, воспроизводить потоковый контент из Интернета.

Музыку и видео можно добавлять к вашей настройке или с помощью устройства USB, или



➤ Это дистанционное управление по умолчанию; в настройках RaspBMC доступны варианты с более привлекательной графикой.

прямо на SD-карту с помощью FTP (имя пользователя – pi, пароль – raspberry). См. ниже информацию по поиску IP-адреса.

Вероятно, единственная настройка, которую вам придется сделать – это позаботиться о том, чтобы звук направлялся в нужное место. В System > System > Audio Output проверьте, чтобы Output был установлен на Analog, если вы используете гнездо, или HDMI, если используете этот интерфейс.

Все это хорошо, но управлять телевизором мышью и клавиатурой как-то не очень комфортно. Славные парни из XBMC учли это, и добавили поддержку дистанционного пульта. Самый простой и удобный способ сделать это – используя web-интерфейс RaspBMC. С его помощью любое устройство с web-браузером, находящееся в той же сети, что и Pi, может управлять воспроизведением.

Эта функция включена по умолчанию, и все, что от вас требуется – это выяснить IP-адрес Pi в System > System Info (нужно настроить его так, чтобы он был статическим IP на вашем маршрутизаторе). Затем на любом другом устройстве, присоединенном к сети, откройте браузер и направьте его на этот IP.

Те, кто идет в ногу со временем и используют мобильное устройство с Android или iOS, найдут солидный выбор в разнообразных магазинах приложений, и смогут сделать то же самое в интерфейсе попрятнее, чем страницы HTML.

Мы обнаружили, что Raspbian вытягивает всю энергию из источника питания. Если вы добавите еще и периферийные устройства и обнаружите, что он стал нестабилен, причина может быть именно в этом. Лучшим способом решения данной проблемы будет подключение всех устройств USB через хаб с собственной подпиткой.

Пойдем еще дальше

Можно полностью управлять ТВ с помощью Linux, например, смотреть прямые трансляции и записывать передачи – например, при посредстве *MythTV* (который находится на www.mythtv.org). Вам понадобится отдельный компьютер с соответствующими кабельными соединениями, который работал бы в качестве сервера.

Однако мы должны предупредить: *MythTV* известен привередливостью своей настройкой. Из-за стрессов, полученных во время этой процедуры, у редактора *LXF* Грэма Моррисона появился не один седой волос.

Вы можете воспроизводить видеофайлы, хранящиеся на других компьютерах в вашей сети, например, на сетевом устройстве хранения (Network Attached Storage, NAS). Выбор способа воспроизведения зависит от типа выдачи доступа к этим файлам, но тот настраивается кнопками Add Sources. Более подробную информацию вы найдете в wiki на сайте <http://bit.ly/00vXb6>.



➤ Новые платы revision 2 имеют монтажные отверстия, позволяющие придать более аккуратный вид вашему развлекательному центру.

Контроллер камеры



Создайте резервные копии своих фотографий с помощью Pi.

Размер Raspberry Pi позволяет управлять им другими встраиваемыми устройствами. Это может показаться перебором – ведь встраиваемые устройства в той или иной форме уже снабжены контроллерами; но это означает, что можно расширять их и создавать для них скрипты такими способами, которые просто невозможны (или, по крайней мере, очень трудны) без дополнительных устройств. Практически все, что подключается к обычному настольному компьютеру, можно запрограммировать на Pi, но мы здесь рассмотрим камеры, на что есть несколько причин. Во-первых, большинство их поддерживается в Linux, и во-вторых, есть ряд полезных проектов, которыми вы сможете заняться, разобравшись с основами.

Лучший инструмент командной строки для работы с камерами в Linux – *Gphoto2*. Его можно установить командой `apt-get install gphoto2`

Прежде чем углубиться в проект, рассмотрим, что умеет этот полезный инструмент. Среды рабочего стола может попытаться подмонтировать камеру, из-за чего у *Gphoto2* бывают проблемы, так что самый простой способ – запустить его без этого. Откройте терминал и запустите `sudo raspi-conf`, и в Boot Behaviour выберите No, чтобы не запускать систему окон, и затем перезагрузитесь. В нашей тестовой

Подкормите свой Pi

Raspberry Pi подпитывается от своего порта micro-USB. Это дает ему 5 В, а Raspberry Pi Foundation рекомендует ток не менее 700 мА, получаемый от сетевого адаптера или через кабель USB от компьютера.

Если вы заинтересованы в портативности вашего Pi, есть другие возможности. Четыре бата-

рейки AA обеспечат ему достаточное питание, при условии, что у вас есть соответствующее гнездо и кабели для питания порта micro-USB.

Однако мы пришли к выводу, что наилучшим решением будет применить шнур питания для мобильного телефона, подключив его непосредственно к Pi.

системе мы обнаружили, что таким образом можно запустить все, что угодно, обойдясь энергоснабжением Pi, но если в то же время использовать мышь, придется применить

«Размер Pi позволяет управлять встраиваемыми устройствами.»

подпитываемый хаб. Естественно, это зависит от особенностей ваших периферийных устройств и энергоснабжения.

В новой текстовой среде подключите камеру и скомандуйте

```
gphoto2 --auto-detect
```

Команда попытается найти все камеры, присоединенные к вашему Pi. Будем надеяться,

она и вашу обнаружит. И хотя она поддерживает внушительный список камер, есть несколько таких, которые работать не будут. Если ваша относится как раз к числу неудачников, то вам

придется выпросить, стырить или занять камеру у друга, прежде чем продолжать процесс.

Не все поддерживаемые камеры одинаковы, и следующим шагом будет выяснение того, что умеет камера. Для вывода

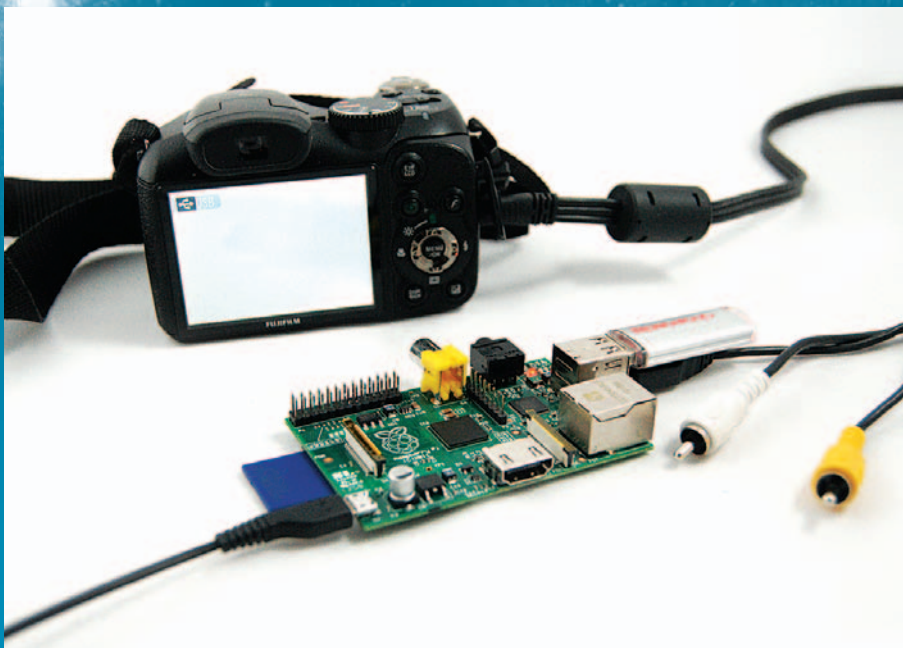
списка имеющихся действий запустите `gphoto2 --auto-detect --abilities`

Не вдаваясь в подробности, можно сказать, что есть два класса способностей: съемка и загрузка/скачивание. Первая позволяет делать снимки с помощью скриптов и присутствует в основном на высококачественных камерах. Вторая позволяет работать с фотографиями, хранящимися на карте памяти, и имеется в большинстве поддерживаемых камер. В данном проекте мы будем работать только со вторым набором способностей.

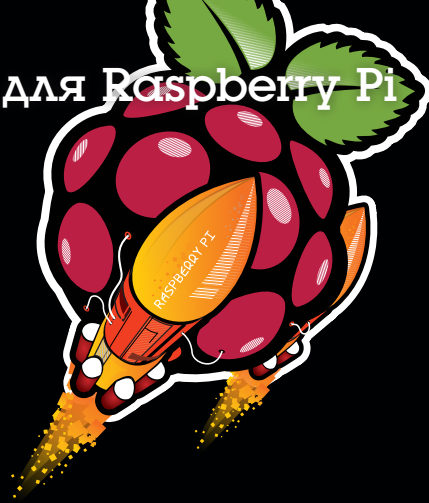
Самая простая команда, которую мы можем отправить в камеру – получить все фотографии, хранящиеся в ней:

```
gphoto2 --auto-detect --get-all-files
```

Она запустит скачивание всех фотографий с камеры в текущую директорию. На обычном компьютере это нормально, но вам, возможно, не захочется делать этого на Pi, поскольку есть риск слишком быстро заполнить память. Вместо этого мы скопируем их на флэшку USB. Чтобы сделать это в интерактивной сессии, можете просто подмонтировать флэшку инструментом GUI, затем запустить `df -h`, чтобы увидеть, куда она подмонтирована, и перейти в эту директорию. Однако, поскольку это будет происходить автоматически, нам надо знать, где будет находиться устройство. Это можно сделать несколькими способами, но мы не будем усложнять. Мы подмонтируем первый раздел первого серийного диска, и будем сохранять фотографии туда.



В таком виде он не слишком портативен, но, применив смекалку и умение, вы сможете сделать его удобнее.



Мы предполагаем, что вы работаете как пользователь по умолчанию — pi. Если нет, надо настроить скрипт. Во-первых, нам нужна точка монтирования для диска. Это просто папка, и ее можно воткнуть куда угодно — мы «нарушим конвенцию» и поместим ее в нашу домашнюю директорию. Поэтому перед вызовом скрипта запустите

```
mkdir /home/pi/pic_mount
```

Теперь мы готовы приступить. Вот скрипт для монтирования диска и скачивания фотографий:

```
#!/bin/bash
if mount /dev/sda1 /home/pi/pic_mount ; then
echo "Раздел смонтирован"
cd /home/pi/pic_mount
yes 'n' | gphoto2 --auto-detect --get-all-files
umount /dev/sda1
else
echo "/dev/sda1 невозможно смонтировать"
fi
```

yes 'n' — это команда, которая просто выпускает поток символов **n**. Это означает, что команда *Gphoto2* переписать ранее загруженные файлы будет отклонена. Крайне важна **umount**, потому что она обеспечивает должную синхронизацию диска и возможность его удаления.

Мы назвали скрипт **get-pics.sh**, и сохранили его в домашней директории Pi. Чтобы сделать его исполняемым, запустите

```
chmod +x /home/pi/get-pics.sh
```

Теперь вы должны суметь запустить его вручную. Вам придется использовать **sudo**, потому что скрипту нужно монтировать диск.

Последний кусочек мозаики — заставить скрипт запускаться автоматически. Для этого мы добавили его в файл **/etc/rc.local**. Этот скрипт запускается при загрузке и выполняется от имени **root**, так что насчет разрешений переживать незачем. Просто откройте файл в текстовом редакторе от имени **root**, например, **sudo nano /etc/rc.local**, и добавьте строку **/home/pi/get-pics.sh**

```
///end code///
```

прямо перед строкой **exit 0**.

Вам остается только подключить свою камеру (убедитесь, что она включена) и устройство USB, и ваши фотографии будут копироваться при загрузке.

Пойдем еще дальше

Если вы хотите запустить устройство без клавиатуры и мыши, что вполне ожидаемо, вы можете присоединить светодиоды (LED) к контактам GPIO, как это было показано в нашей статье, и использовать их для отображения статусов.

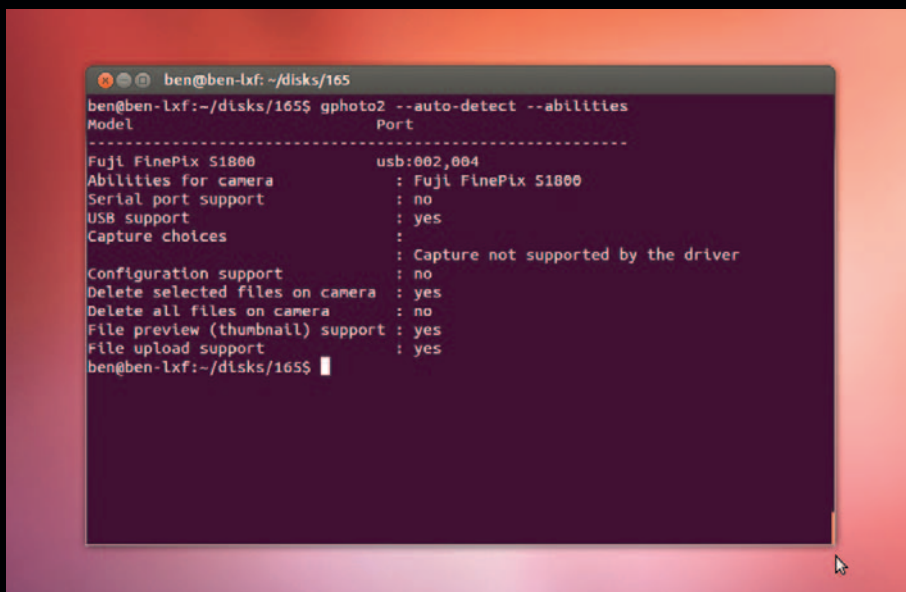
Помимо сохранения фото на устройстве USB, можете загружать их на онлайн-сервисы — например, Flickr. В таблице по беспроводному подключению вы найдете информацию о том, как подключить ваш Pi к телефону.

Вы можете ввести нечто вроде выключателя, чтобы сообщать Pi, какие фотографии нужно загрузить, а какие сохранить на USB-носителе: например, загружать изображения с низким разрешением и сохранять изображения с высоким разрешением. Или можно

создать версии изображений с низким разрешением и загрузить их.

Конечно, на этом можно и не останавливаться. Если в вашем Pi имеется сетевой адаптер, воспользуйтесь им для запуска сервера HTTP. С помощью скриптов PHP (или другого языка) вы сможете создать интерфейс для *Gphoto2*, который позволит вам подключаться к своему мобильному телефону.

Теперь развернемся в другую сторону: если ваша камера поддерживает опции съемки, вы сможете не только копировать своим Pi фотографии, но и делать их.



» *Gphoto2* умеет намного больше, чем мы здесь попробовали, в том числе связки для Java и Python. Чтобы узнать о них все, загляните на сайт www.gphoto.org.

Работа в сети

Raspberry Pi поставляется с проводным соединением Ethernet, что прекрасно подходит в большинстве случаев, но иногда кабель просто не дотягивается до нужного места.

При желании вы можете использовать беспроводной USB-брелок. Однако если у вас есть телефон Android и ваш носитель не отключил эту функцию, примените его в качестве сетевого устройства. У него имеется то преимущество, что он потребляет меньше энергии, чем Pi, и поэтому облегчает возможность употребления для его питания батареек.

У вас должно получиться подключить телефон и к Wi-Fi, и к 3G, чтобы не повлиять на лимит данных. Конечно, лучше всего проверить тип подключения до начала закачки больших файлов. Для этого подключите телефон к Pi, и отключите режим модема [tethering] в Settings > Wireless and Networks > Tethering and Portable Hotspots (на телефоне). В Pi, если вы введете **sudo ifconfig**, вы должны увидеть список интерфейса **usb0**, но у него не будет IP-адреса.

Сетевые интерфейсы управляются файлом **/etc/network/interfaces**. По умолчанию здесь нет пункта для сетевого соединения USB, и нам надо

его настроить. Откройте файл своим любимым текстовым редактором через **sudo** — например, с помощью **sudo nano /etc/network/interfaces** — и добавьте строки

```
iface usb0 inet dhcp
nameserver 208.67.220.220
nameserver 208.67.222.222
```

Здесь мы использовали серверы имен OpenDNS, но вы можете при желании использовать другие.

Теперь можете перезагрузить интерфейс или Pi, чтобы изменения вступили в силу, и у вас должно появиться рабочее интернет-соединение.

Передача информации

Используйте контакты GPIO, чтобы зажечь несколько светодиодов (LED).

Благодаря малому размеру, Raspberry Pi идеален для создания собственных встраиваемых устройств. Так можно успешно создавать небольшие компьютерные устройства для решения специфических проблем, что мы продемонстрировали с контроллером камеры. Однако есть небольшая загвоздка: как узнать, что происходит внутри вашего Pi, без экрана? К счастью, разработчики Pi подумали об этом и добавили возможность обмениваться информацией с Pi без множества обычных периферийных устройств для ПК. Это делается с помощью General Purpose Input and Output (GPIO) [Ввод

и Вывод Общего Назначения]. Возможно, вы недоумевали, зачем нужны эти острые штырьки около устройства считывания SD-карты – что ж, теперь вы знаете. Эта базовая схема может использоваться для отображения информации из любого источника, но здесь мы используем ее, чтобы решить проблему, с которой часто сталкиваемся в редакции LXF: обнаружение конечного байта IP-адреса.

Это полезно, если вам нужно удаленно управлять доступом к своему Pi, но нельзя настроить его статический IP-адрес, потому что, например, он перемещается из одной сети в другую. Обычно первые три байта вычис-

лимы по сетевой маске, но последний может от вас ускользнуть, если у вас нет монитора.

Мы будем использовать программу *gpio*, являющуюся частью WiringPi. Более подробную информацию о ней ищите на сайте <http://bit.ly/RP8UKJ>.

Эта программа имеется также на DVD. Она идет в виде исходного кода, так что нам нужно распаковать его и скомпилировать с помощью

```
tar xvf wiringPi.tgz
cd wiringPi/wiringPi
make
sudo make install
cd ../gpio
make
sudo make install
```

Также мы будем использовать *bc*, поэтому установите его, командой

```
sudo apt-get install bc
```

Ну, довольно программ – перейдем к железу! Но сначала предупреждаем: неправильно соединив провода, вы сломаете свой Pi, так что дважды все проверьте перед тем, как включить питание.

Схема для этого очень проста: нужно просто соединить каждый вывод с плюсовой ножкой светодиода, потом минусовую ножку светодиода (обычно более короткую) – с резистором 1 КОм, и, наконец, другой контакт резистора – с общей землей. Более подробно это показано на рис. 1, 2 и 3.

Соединив полностью настроенную плату с Pi, приступим к работе. Для начала просто используем конечный контакт. Это контакт 7 (контакты расположены не в порядке номеров). Откройте терминал и настройте его на вывод командой

```
gpio -g mode 7 out
```

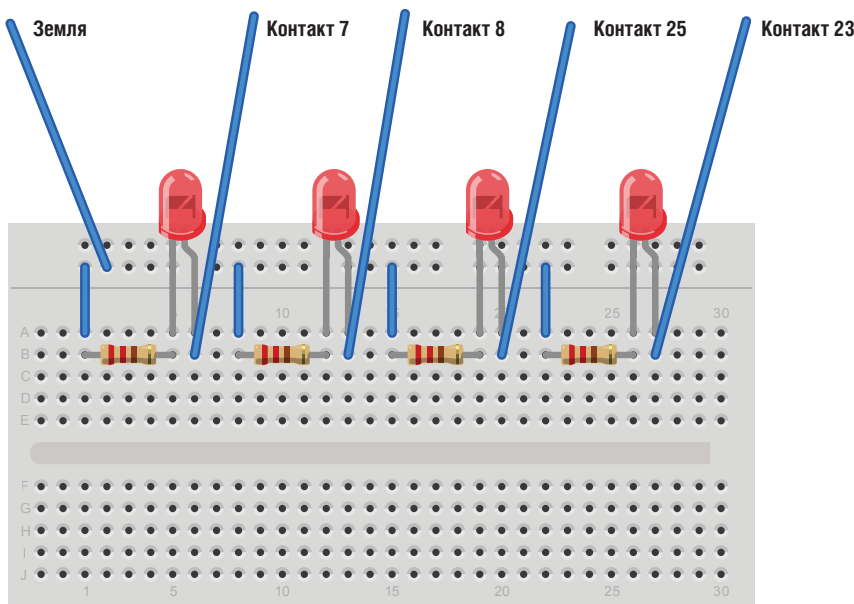


Рис. 1. Здесь показано, как соединена половина светодиодов. Все дополнительные подключаются точно так же.

Закон Ома

Есть два основных способа измерить электричество: напряжение и сила тока. Напряжение (измеряется в вольтах) – это количество энергии, которым располагает определенное число электронов, а сила тока (измеряется в амперах) – это количество электронов, проходящих через определенную точку.

И оба этих параметра связаны между собой законом Ома, который гласит: Напряжение=Сила тока*Сопротивление, или $U=IR$. Вам следует учитывать эту взаимосвязь, чтобы случайно не спалить свой Pi, пропустив через него слишком сильный ток.

Настройка Pi представляет собой некоторую сложность. Для желающих как следует разобраться с ней Герт ван Лоо [Gert van Loo], один из разработчиков, составил объяснение, которое можно найти здесь: <http://bit.ly/Qp4PMI>.

Если рассматривать все это с практической точки зрения, вы можете рассчитывать на напряжение на выходном контакте GPIO величиной 3,3 В, и вам нельзя превышать силу тока 16 мА или подавать большую величину на входной контакт.

Это максимальная величина силы тока; старайтесь использовать ток меньшей силы. Итак,

поскольку нам известно, что по закону Ома $U=IR$, то $R=U/I$. Если мы получаем данные с Pi и хотим быть уверенными в их сохранности, мы знаем, что R должно быть больше, чем 3,3/0,016, что составляет 206,25 Ом.

Помните, что это – самая малая величина сопротивления, которую можно безопасно использовать с выходом GPIO. Вам надо стремиться к обеспечению уровня безопасности в несколько раз выше этого, если только не возникнет крайней необходимости. В наших схемах мы использовали килоомы, что обеспечило нам фактор безопасности почти 5.



Затем включайте его с помощью

```
gpio --g write 7 1
а отключайте –
gpio --g write 7 0
```

Если вы похожи на нас, то вы будете заниматься этим, пока не надоест. После чего вы созрели запустить скрипт. Он разбит на четыре части. Первая часть просто настраивает контакты на нужный режим и обеспечивает их выключение:

```
pins="7 8 25 24 23 18 15 14"
for x in $pins
do
  gpio -g mode $x out
  gpio -g write $x 0
done
```

Вторая получает IP-адрес от `ifconfig`, превращает его в двоичный, затем при необходимости снабжает нулями в начале.

```
ipaddress=`ifconfig eth0 | grep 'inet ' | awk
'print $2' | cut -f4 -d'.'`
binary=`echo "ibase=10;obase=2;$ipaddress"
| bc`
paddedBinary=`printf %08d $binary`
```

Следующая часть посредством `cut` вырезает нужную нам часть из этой двоичной строковой переменной и выводит ее на соответствующий контакт.

```
bit=1
for x in $pins
do
  out=`echo $paddedBinary | cut -b$bit`
  gpio -g write $x $out
  bit=$((bit+1))
done
```

И, наконец, мы даем скрипту команду поспать в течение пяти минут, а затем отключаем светодиоды.

```
sleep 5m
for x in $pins
do
  gpio -g write $x 0
done
```

Вот и все! Скопируйте `showIP.sh` с DVD, сделайте его исполняемым с помощью `chmod a+x showIP.sh`

Gertboard и Arduino

Подключение напрямую к контактам GPIO вашего Pi может предоставить вам базовый контроль ввода-вывода, но оно не лишено ограничений. Есть два дополнительных компонента, которые помогут вам более точно взаимодействовать с миром вокруг вас.

Достаточно полный комплект расширений для связывания Pi с реальным миром – Gertboard, включающий микроконтроллер, а также ряд опций ввода-вывода. Он поставляется в разобранном

виде, и вам придется поработать паяльником, чтобы его собрать.

А вот Arduino – это микроконтроллер, подключаемый к Pi (или любому другому компьютеру) через порт USB. Как правило, он поставляется готовым, но доступны также и комплекты частей. В сыром виде он имеет меньше функций, чем Gertboard (который включает микроконтроллер Arduino), однако может быть расширен огромным диапазоном щитков.

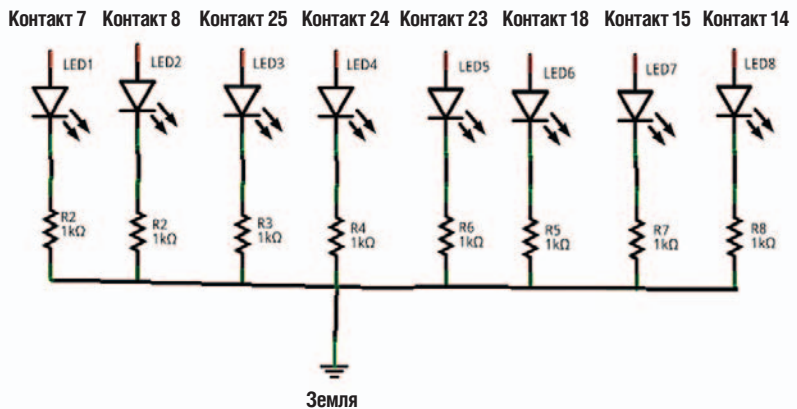


Рис. 3. Простая схема во всем блеске.

и введите `sudo ./showIP.sh` для отображения своего IP-адреса. Чтобы все это делалось автоматически при загрузке, добавьте строку `/home/pi/showIP.sh &`

в `rc.local`. В разделе Контроллер Камеры вы найдете более подробную информацию о том, как это сделать.

Мы показали вам, как передавать информацию через GPIO, но его название предполагает, что оно может также и принимать информацию.

При этом еще более важно не направить слишком сильный ток на контакты. Чтобы при-

нять информацию, просто настройте режим на прием с помощью `gpio --g mode <pin number> in`, затем считайте показатель с помощью `gpio --g read <pin number>`.

Это оборудование может отображать восемь битов любой информации, и вы можете не ограничиваться отображением только IP-адресов. Например, можно сделать модифицированную версию скрипта контроллера камеры, используя светодиоды для отображения его состояния.

Более подробную информацию по всем контактам GPIO вы найдете на <http://bit.ly/JTIFE3>. Контакты, которые мы употребили, одинаковы в версиях 1 и 2 Raspberry Pi, но некоторые контакты впоследствии изменились. Если вы создаете собственную схему или используете схемы, найденные в Сети, убедитесь, что контакты пригодны для вашей платы.

Не ограничивайтесь простым включением и выключением контактов. Pi поддерживает несколько способов передачи более солидных объемов данных через GPIO. Самые популярные – шина Serial Peripheral Interface (SPI) и Inter-Integrated Circuit (I2C).

Есть ряд устройств, которые их используют, и масса информации, которая поможет вам разобраться. Итак, чего же вы ждете? Берите паяльник – и вперед, к созданию собственной армии роботов. LXF

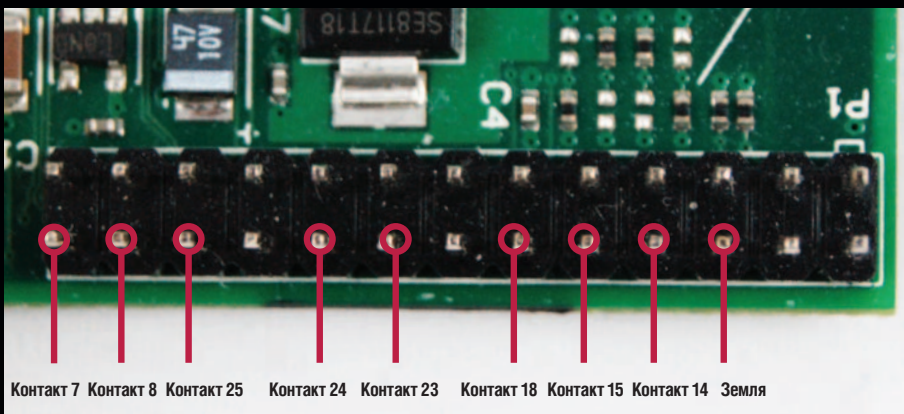


Рис. 2. Подключите к этим контактам макетную плату. Мы использовали готовые одноконтактные разъемы, но вы можете припаять разъемы или использовать старый кабель IDE.

Accessibility: Взгляд изнутри



Михаил Пожидаев анализирует концепции и перспективы технологий доступа к компьютеру для слабовидящих.



Наш эксперт

Михаил Пожидаев – инженер-программист компании «Альт Линукс», кандидат технических наук, преподает в Томском государственном университете. В свободное время увлекается игрой на фортепиано и чтением книг.

В наше время представить себе полноценную жизнь без компьютера практически невозможно. И, естественно, Accessibility, то есть облегчение доступа к компьютеру пользователям, имеющим физические ограничения, является не последней заботой также и операционной системы GNU/Linux, о чем рассказывалось в **LXF146**. В частности, приводилось описание функций утилиты чтения экрана *Orca*, предназначенной для людей с проблемами зрения. Ниже мы предлагаем читателю более детально познакомиться с компонентами, лежащими в основе вспомогательных технологий GNU/Linux, а также проанализировать возможное развитие этого направления в будущем.

Экранный тчец *Orca* разработан для представления оконного интерфейса Gnome в речевом виде. При каждом действии пользователя он озвучивает элемент, на котором остановился фокус ввода. Не глядя использовать указатель мыши обычно достаточно тяжело, поэтому все команды передаются при помощи клавиатуры. Основная работа по организации доступной среды выполняется двумя крупными сервисами, взаимодействие которых *Orca* поддерживает с учетом параметров, установленных пользователем.

Один из этих сервисов – так называемый речевой сервер. Он занимается обслуживанием речевых синтезаторов, выполняет их запуск и организует воспроизведение звукового потока.

Термин «сервер» правильно отражает назначение приложения, поскольку возможно существование нескольких источников речи (клиентов), для которых необходимо правильно организовать совместную работу. Тем не менее, речевой сервер не является сервером в полном смысле этого слова, потому что клиенты не изолированы друг от друга, допускается воспроизведение только одного фрагмента текста в каждый момент времени, и клиенты вправе отменять выполнение заданий друг друга. Кроме *Orca*, источниками речи могут быть различные системные оповещения,

включающие уведомления о подключении съемных носителей информации, речевые часы и пр.

Другой сервис, называемый *AT-SPI*, играет более важную роль в процессе предостав-

«Утилита Orca предназначена для людей с проблемами зрения.»

ления вспомогательных технологий. Название расшифровывается как Assistive Technology Service Provider Interface [Интерфейс поставщика сервиса вспомогательной технологии]. *AT-SPI* занимается сбором полной информации о пользовательском интерфейсе приложений, включая сведения об открытых окнах, расположении элементов управления и других деталях, позволяющих получить представление об изображении на экране. Реализация *AT-SPI* подразумевает решение двух крупных технических задач: внедрение в библиотеки построения оконного интерфейса компонента для получения необходимой информации и сбор полученной информации в рамках одного процесса для последующей трансляции в речевую форму. Для решения первой задачи требуется подготовить дополнения ко всем популярным оконным библиотекам, а также ко всем приложениям, использующим собственные реализации оконных виджетов. Поскольку развитие *AT-SPI* велось преимущественно в рамках Gnome Accessibility Project, то первой библиотекой, получившей необходимую поддержку, стала *GTK+*. Со временем подобные дополнения появились в браузере *Firefox*, офисном пакете *OpenOffice.org* (а впоследствии и в *LibreOffice*), в *Java/Swing* и, наконец, в *Qt4*. Таким образом, вспомогательные технологии будут доступны в большинстве приложений, получивших распространение в GNU/Linux.

Для централизованной обработки полученной информации используется специальное приложение *at-spi-registryd* (см. схему справа), запускаемое при старте оконной системы и функционирующее в фоновом режиме. Основная сложность в реализации концепции *AT-SPI* заключается в том, что *at-spi-registryd* и компоненты, интегрированные в оконные библиотеки, исполняются в разных процессах. Обмен информацией напрямую невозможен, и применяются средства организации межпроцессного взаимодействия. Первая версия *AT-SPI* использовала в своей работе *Corba/VoPobo*, но в дальнейшем от этого выбора пришлось отказаться, поскольку в Gnome 3 использование *Corba/VoPobo*



➤ Дельфин-косатка (лат. *Orcinus orca*) окраской напоминает нашего любимого пингвина.

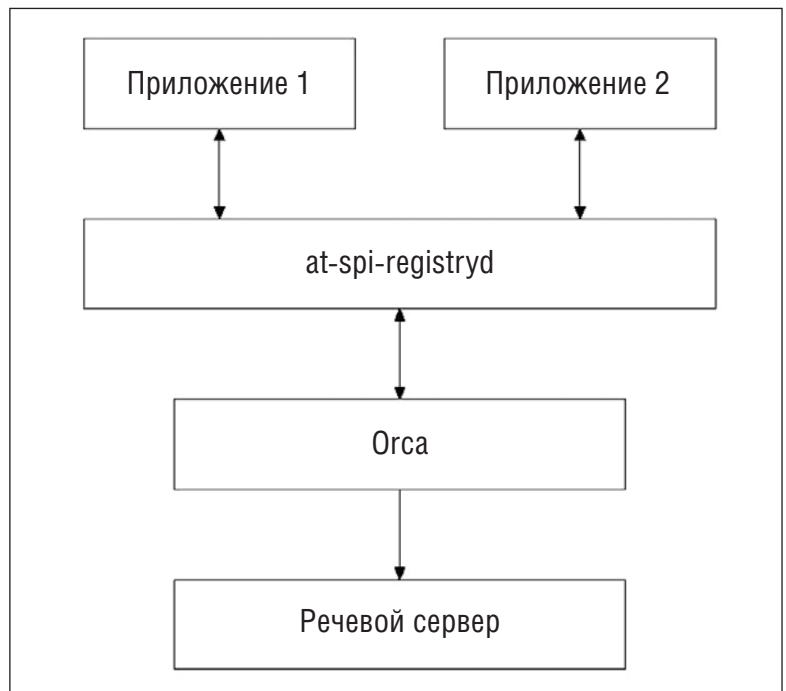
было признано устаревшим, и на его замену пришел сервис D-Bus. Обычно при использовании D-Bus задействуется одна из двух так называемых шин: либо системная шина, функционирующая для обмена информацией среди всех приложений ОС, либо сессионная шина, при помощи которой выполняется взаимодействие приложений внутри одной сессии пользователя. Для работы AT-SPI был выбран третий подход, заключающийся в запуске отдельной копии сервера D-Bus, предназначенной для обслуживания взаимодействия приложений только в рамках задач AT-SPI. Такой выбор призван повысить производительность системы, поскольку проблемы скорости работы AT-SPI очень актуальны: информации об интерфейсе приложений очень много, и передача через D-Bus занимает значительное время. В завершение описания концепции AT-SPI заметим, что утилита *ChromeVox*, предлагаемая компанией Google для работы незрячих и слабовидящих людей в браузере *Chrome*, не использует функций AT-SPI, хотя это не исключает появление поддержки в будущем.

Концепция AT-SPI, обещающая достаточно широкие возможности для работы незрячих и слабовидящих людей с оконными приложениями, продумана очень тщательно, но существуют и альтернативные подходы. Например, интересный вариант основан на добавлении вспомогательных функций в текстовый редактор *GNU Emacs*. Как известно, круг задач, решаемых при помощи *GNU Emacs*, не ограничивается текстовым редактором. За счет многочисленных дополнений пользователи имеют возможность работать с электронной почтой, с удобным файловым менеджером, составлять записи в дневнике, просматривать календарь, вести разработку ПО и пр. Доступен упрощенный вариант web-браузера, основным ограничением которого является невозможность исполнения Java Script. Дополнения к *GNU Emacs* подготовлены таким образом, что все рабочие объекты представлены в текстовом виде. Именно это сыграло решающую роль в популярности подхода, основанного на *GNU Emacs*.

Необходимый набор вспомогательных функций для *GNU Emacs* поставляется в пакете *emacspeak*. Основная задача *emacspeak* – выполнять речевые оповещения в ответ на все действия пользователя. При перемещении курсора вверх-вниз *emacspeak* произносит полный текст строки, в случае движения вправо-влево – символ в новом положении курсора. Рабочее пространство может разбиваться на несколько окон, в каждом из которых отображается некоторый документ. Помимо приложений из состава самой среды *GNU Emacs*, пользователь может использовать утилиты командной строки, запуск которых выполняется при помощи терминала, оснащенного речевым интерфейсом.

Речевой вывод, формируемый *emacspeak*, перенаправляется в речевой сервер, функции которого совпадают с функциями речевого сервера, используемого для работы утилиты *Orca*, упомянутой выше.

Основное преимущество работы в *GNU Emacs* заключается в высокой скорости взаимодействия с ПК незрячего или слабовидящего человека, которая достигается за счет отсутствия необходимости навигации по элементам оконного интерфейса, ориентированного изначально на представление информации в графическом виде и управление при помощи мыши. Важную роль в достижении высокой скорости играют и широкие возможности назначения «горячих» клавиш. Гармоничным дополнением к среде *GNU Emacs* в случае использования незрячим или слабовидящим человеком является доступность таких пакетов, как *LaTeX* (со входящими в него *MetaPost* и *Beamer*), *Lilypond* и *GNUPlot*. Все эти пакеты объединяет возможность подготовки материалов



в текстовом виде с последующим экспортом в PDF-файл. Обладая необходимыми знаниями и навыками, пользователь может работать над большими текстами (книгами, диссертациями), нотными партитурами и презентациями. Работа производится незрячим человеком в основном самостоятельно, но последующая визуальная проверка остается необходимой.

В сообществе GNU/Linux накоплено много опыта работы при помощи речевого интерфейса – как с использованием оконной среды, так и в *GNU Emacs*. AT-SPI имеет качественную реализацию и может стать стандартом вспомогательных технологий для всех оконных окружений, а не только для Gnome. Поддержка AT-SPI обсуждается в сообществах *Xfce*, *KDE* и даже *Tizen*, но в каждом случае требуется подготовка новой утилиты на замену *Orca*, которая имеет жесткие зависимости с библиотеками Gnome. Главная проблема этого направления в том, что разработчики приложений нередко прибегают к использованию

собственных элементов управления, забывая о необходимости совместимости со вспомогательными технологиями. При работе с *GNU Emacs* тоже существуют трудности. Основная проблема заключается

в том, что *GNU Emacs* изначально ориентирован на опытного пользователя, уровень знаний которого приближается к уровню разработчика ПО. Пользовательские настройки сохраняются в виде скриптов на языке Lisp, а многие задачи требуют подготовки сценариев *Bash*. Ситуацию усложняет невозможность перевода речевого вывода на национальные языки, это делает массовое распространение очень затруднительным, но приведенные недостатки не вытекают из самой концепции представления рабочих объектов в текстовом виде, которая позволила выработать подход к работе незрячего или слабовидящего человека на ПК со скоростью, сравнимой со средней скоростью работы людей в различных коллективах. Дальнейшее развитие этого направления дает надежду на появление новых продуктов, лучше приспособленных для использования в рамках учебного или производственного процесса, чем подходы с озвучиванием оконного интерфейса. Присоединяйтесь! **LXF**

► **Потоки передачи вспомогательной информации в оконной системе.**

«Накоплено много опыта работы при помощи речевого интерфейса.»

Полнейший Монти

LXF махнул в Портленд на встречу с со-творцом MySQL — человеком, добавившим 'M' в LAMP.



Монти Видениус (Monty Widenius), один из создателей СУБД MySQL, стал мультимиллионером, когда в 2008 ее купили Sun

Microsystems. Но затем, незадолго до перехода Sun под Oracle, сам же покинул MySQL, уведя за собой многих первоначальных разработчиков на свое ответвление, MariaDB. Мы встретились с ним в Портленде, чтобы узнать, почему важна философия свободного ПО, что случилось в Sun и почему в истории MySQL интриг и страстей больше, чем в первом сезоне «Далласа».

LXF: Мы всегда считали, что с базами данных иметь дело труднее всего.

МВ: Не спорю, это первое, что пришло мне в голову, когда я с ними связался. Я их ненавидел.

LXF: Что же изменилось? И как это захватило вас так надолго?

МВ: На самом деле, все началось в 1981-м — я тогда работал в компании, где для работы с данными использовалась версия BASIC. Меня попросили перевести их на TRS-80, с помощью TRS-80 BASIC, и при этом никакой базы данных, никакого хранилища. Дело было перед моим поступлением в колледж, и я ломал голову, как сохранить данные на диск, чтобы можно было ими пользоваться. Затем я придумал такой способ — как выяснилось в колледже, это называлось хэшинг, но я тогда этого не знал. По сути, на нем мы основали целую платежную систему.

LXF: Так вы, стало быть, самоучка?

МВ: Да, во всем. То есть я, конечно, учил информатику в колледже, пару алгоритмов там выучил, а потом выяснилось, что до хэшинга я дошел сам. В основном, я предпочитал сам исследовать.

LXF: Но когда это перешло в контекст философии открытого ПО?

МВ: С моим приходом в UNIX — в 1984-м, кажется. В 1985 были кризисы между Швецией и Финляндией, на одном из них я познакомился с Дэвидом Аксмарком [David Axmark], другим создателем MySQL. Мы уже поработали с Emacs и присматривались к GCC, и нас как-то зацепило. Я стал работать с компаниями, использующими Solaris. И обнаружил, что все его оболочки довольно неуклюжи.

Вот мы и заменили все это на Bash, GCC и прочее. Так я и попал в Open Source, и когда мы все чаще стали ездить на посвящен-

О БЫЛЫХ ВРЕМЕНАХ

«Люди побаивались выпускать код, ведь любой мог его присвоить.»

ные ему конференции, почти каждый год, нам с Дэвидом захотелось внести свою лепту. Однако ничего полезного широкой аудитории у нас не было, и мы не чувствовали, что сможем обеспечить поддержку. А создавать ПО без поддержки — значит увеличивать го-

ру бесполезного, чем и так многие занимаются; и мы стали искать такое, над чем могли бы позволить себе работать с полной занятостью.

А в 1994-м, на базе старого и опять-таки бесполезного проекта я создал MySQL, и мы отметили, что польза в нем есть. Тогда мы его выпустили на волю — и он взлетел.

LXF: Тогда вы и поняли впервые, что лицензия Open Source пойдет на пользу проекту?

МВ: Мы были консалтинговой компанией и рассудили так, что, по крайней мере, этот выпуск не принесет нам убытков благодаря нашей основной деятельности. Если кто-то и возьмет программу, то все равно наймет нас консультантами, поскольку она большая и сложная. Так что в деньгах мы не потеряем.

LXF: Все же многие рассудили бы иначе.

МВ: Ну, тогда люди побаивались выпускать код, ведь любой мог его присвоить. А мы выпустили MySQL под двойной лицензией, обеспечив, что если кто-то захочет бесплатно применять его дома, все в порядке, мы не против. Но вот заработать на ней помимо нас не получится.

LXF: Вы не использовали GPL?

МВ: GPL уже существовала, но мы ею не пользовались, потому что для маленькой компании это был большой риск; так что у нас была лицензия, разрешающая использовать исходный код для чего угодно, но если хочешь с этого зарабатывать — свяжись с нами и обговори условия. Или покупай коммерческую лицензию — и делай, что хочешь.



LXF: И как, стали люди вас обзывать, говоря, что желают коммерции?

МВ: Да. В середине 1998-го мы сделали лицензию более свободной. У нас была версия для Windows – вернее, тогда только она и была, но мы решили, что Linux будет доступнее. По версии для Windows мы говорили, что это условно-бесплатная лицензия. Условно-бесплатная программа [shareware]. После месяца пользования нужно было заплатить \$200 или жить с нечистой совестью. На свой выбор. Была еще web-страница, где можно было зарегистрироваться, а затем отправить по факсу \$200, и наш факс работал непрерывно.

На этой стратегии наш штат к 2000 году вырос от 2 до 15 сотрудников только за счет Windows-версии. А в 1999-м меня завалили жалобы от наших клиентов, почему это мы не используем GPL. И тогда мы решили, что по деньгам уже можем позволить себе рискнуть.

LXF: Вы все еще считали, что GPL — это риск?

МВ: Да, ведь до этого все, кто хотел зарабатывать деньги с нашим ПО, обращались к нам. Выпустив MySQL под GPL, мы не знали, что теперь это можно будет делать и в обход. И за первые два месяца наши доходы действительно упали более чем вдвое, но за два последующих мы вышли на прежние показатели. Все благодаря тому, что у нас уже была своя большая клиентура, и все были настолько рады появлению GPL, что пользователей у нас только прибавилось. Но на начальной стадии это вряд ли удалось бы.

LXF: Сейчас бы вы поступили так же?

МВ: Мы бы снова вернулись к двойной лицензии – нам кажется, это подспорье для бизнеса, потому что вы принимаете эти условия использования ПО на три года. Условия могут быть любыми, это не свободный код. Три года спустя это уже BSD, и вы можете им пользо-

ваться бесплатно. И, значит, свободно распространять. Уровень доверия тот же, что и с Open Source.

LXF: Своего рода вложение умственного капитала?

МВ: Они же не уходят, вы можете по-прежнему исправлять ошибки и прочее. Просто знают, что раз они пользуются этим три года, то надо как-то компенсировать в ответ. Кроме того, соглашение идет на пользу бизнесу, потому что вы остаетесь верны Open Source, но при этом можете объяснить своим пользователям, что вам нужен некий стартовый капитал. По-моему, это разумный компромисс.

LXF: И давно вы инвестируете в открытые проекты?

МВ: Вкладываю или работаю с ними? Ну, в MySQL я вкладываю время, начиная с 1994 года. Получив деньги от Sun, я создал



инвестиционную компанию, где мы спонсируем технические проекты сообществ, и Open Source, и другие; и этим мы занимаемся с 2008 года.

LXF: И как это продвинулось с годами? Идеи стали лучше? Наблюдается ли рост числа людей, готовых инвестировать в общественные проекты?

МВ: Мы – одна из уникальных в своем роде инвестиционных компаний, вкладывающих деньги в общественные проекты, так что идей у нас море. Нам присылают где-то по три предложения за неделю. Но в основном это лишь идеи, и поскольку часть средств мы получаем от Европейского банка, то, как правило, хотим быть уверены, что по крайней мере сможем их вернуть. И мы не можем вкладывать деньги в идеи, ведь наша задача – не создавать сообщество, а способствовать его развитию. Этим я занимался в MySQL. Его я создавал, но это дело небыстрое, и потому мы хотим вкладывать деньги в компании, которые уже зарекомендовали себя. И активно развивающихся среди них немного. Мы нашли 15 компаний, в которые можно инвестировать или рассмотреть такую возможность.

LXF: Что изменилось, когда проект купила Sun? Почему вам не дали просто оставить все, как было, раз оно работало?

МВ: Проблема была в том, что в MySQL были два разных мировоззрения. Многие планировали сделать IPO [первоначальное публичное предложение акций, – прим. пер.] и перейти к производству проприетарной системы. Без IPO или продажи компании они этого не могли, потому что я тоже должен был принять это соглашение.

LXF: То есть в MySQL были сторонники закрытого ПО?

МВ: Да, соглашение дало бы им такую возможность, а они сделали финт и создали MySQL

Money Administrator, проприетарный проект, который не следовало бы им разрешать, но они вынудили совет директоров его одобрить. Меня это не обрадовало.

LXF: Это было до Sun?

МВ: Это было до Sun. Я понимал, что после продажи нам придется продвигать закрытые функции, мне это было не по душе, но если ваши уч-

О РАСПРЕДЕЛЕНИИ БЛАГ

«Мы вкладываемся в общественные проекты, так что идей у нас море.»

редители – это ваши инвесторы, то право голоса у вас только пока компания продается или стоит на IPO. Потом его уже нет. Так что я был счастлив, когда нас купил Sun – надеялся, что разлад в руководстве прекратится или хотя бы уменьшится.

Кроме того, была еще наша команда разработчиков, большинство из которых стояли за Open Source. Им хотелось работать над открытым проектом, который идет на благо всем, а прибыли хватает на зарплату.

Кое-кто в руководстве начал реорганизовывать разработку, стремясь выгнать таких, как я, желающих работать с открытым кодом, чтобы можно было больше сил бросить на закрытый код. Так что эмоционально меня почти загнали в тупик.

Я полагал, что имея огромный опыт в сфере разработки, Sun оставит отдельными разработку и менеджмент, как две независимые структуры, сохранив команду разработчиков, чтобы те применяли свои умения, полученные в Open Source. Но они решили, что все прекрасно, и, вопреки моим ожиданиям, конфликт не разрешился, а лишь усугубился.

LXF: А зачем было Sun покупать MySQL?

МВ: Потому что у MySQL была клиентская база числом в 17–20 тысяч, часть которой совпадала с базой Sun, а часть – нет. Предлагая полный пакет с базой данных, Sun могли привлечь куда больше клиентов, чем раньше, а MySQL, при поддержке Sun, получал доступ в сферы, ранее недоступные, куда у Sun теперь появилась возможность продавать ПО. Они поняли, что, совместив свое оборудование и программное обеспечение с возможностями MySQL, который будет обходиться им всего в \$70 миллионов год, они выиграют гораздо больше. И я считаю, они мыслили верно, они могли такого добиться. Но проблема была в разделении структур, так что это не сработало.

Кроме того, я уверен, что мы помогли бы Sun наладить дела с Open Source: поддерживая множество открытых проектов, им никогда не удавалось на них заработать – и они рассчитывали, что поскольку мы такое умеем, то и их обучим.

LXF: Странно. Нам всегда казалось, что Sun не слишком интересовались Open Source...

МВ: Они хорошо спонсировали проекты и доводили их до конца, но хороших денег на них не делали. Так, они ничего не получили с *OpenOffice.org*, хотя занимались им лет шесть или семь.

LXF: Но ведь он тогда не особо и развивался...

МВ: Да, но в том вина разработчиков, а не Sun. Sun больше беспокоило то, что проект не принесит прибыли, и они считали, что мы сможем их научить. Проблема в том, что они (Sun) начали переговоры с руководством, а те просто заявили им: «Ко всем продуктам надо создавать закрытые дополнения, и тогда деньги будут». В Sun сказали, что у них так Open Source не делают, но стали об этом задумываться; однако затем я и мои соратники пришли на собрание разработчиков Sun и объяснили, почему эти идеи работать не будут, и нам поверили, так что этого не случилось.

Но я хотел помочь Sun заработать с помощью Open Source, отчасти потому я к ним и присоединился. Я сотрудничал с Грегом Пападопулосом [Greg Papadopoulos], начальником их Отдела разработки, чтобы сделать Sun более открытым и активнее взаимодействовать с сообществами, поскольку вместо того, чтобы принять участие в понравившемся проекте, Sun делали свой, думая, что у них это получится лучше, а затем выпускали его как открытый код. А правильнее было подключаться к уже существующему проекту, что мы и старались до них донести. И вроде бы получилось, но потом у них начались финансовые трудности.

Sun вообще была странной компанией, в том смысле, что на верхнем уровне все было прекрасно. А вот уровнем ниже, там каждый лишь охранял свою кочку: каждый отдел выполнял свою норму, зная, что помощи они другому отделу, продажи у тех увеличились бы вдвое, но сами они потереяли бы 10% от своих продаж; вот никто никому





и не помогал. Каждый думал только о себе, и о каком-либо взаимодействии не могло быть и речи, если это угрожало их частными потерями. Таков средний уровень.

А на нижнем, на уровне разработчиков... у Sun была одна из лучших команд из мною виденных. Именно они позволили Sun продержаться так долго. А руководители среднего звена, идущие сразу за первыми лицами – это именно они уничтожили Sun.

LXF: Тогда-то Sun и перешел к Oracle?

МВ: Да, причем Oracle удалось заплатить за это до обидного меньше его реальной стоимости. Но с другой стороны, одним из мотивов Oracle в покупке Sun была MySQL. Они ясно дали понять Евросоюзу, что без MySQL они Sun не купят.

LXF: Oracle хотели заполучить MySQL, просто чтобы им перестали пользоваться?

МВ: Именно. Они хотели сами всем управлять, но, покупая MySQL, они как бы давали согласие Евросоюзу на то, чтобы продолжать работу над ней в течение пяти лет. Они также заявляли, что будут вкладывать в ее развитие порядка \$20 млн в год.

LXF: Но они вовсе не хотят, чтобы MySQL приносила больше прибыли, чем их собственные базы данных.

МВ: Конечно, нет. Но они не ожидали и того, что я сделаю ответвление, поскольку экономически это невыгодно. Я уже потратил €4 млн на поддержку MariaDB без малейших шансов много на этом поиметь. В лучшем случае, если очень повезет, я верну свои вложения, но не больше. Люди так не поступают. Поэтому такого они ожидать не могли. С другой стороны, они старательно создавали видимость работы над MySQL, даже после потери лучших инженеров, у них была еще отличная команда

О ЗАТРАТАХ

«Я уже потратил четыре миллиона евро на поддержку MariaDB.»

InnoDB, команда NDB отвечала за свой раздел MySQL, и тоже все работало, приложениями тоже кто-то занимался. Остальные команды практически ничего не делали. Вдобавок они утратили контроль над исходным кодом, потому что не осталось людей, которые его

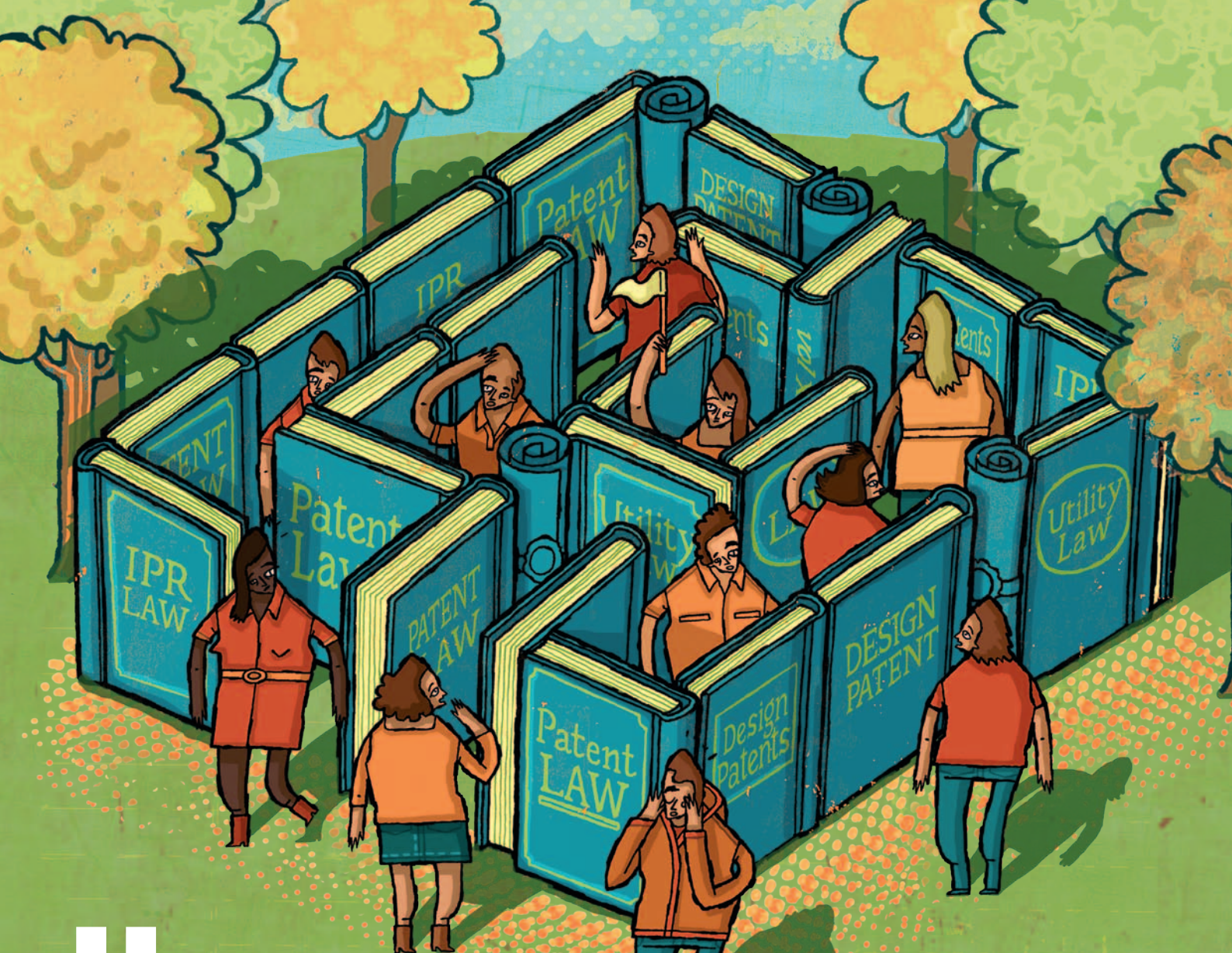
понимали, и поэтому они делают всякие странные вещи, доставляющие нам кучу неприятностей при слиянии программ, потому что мы не можем принять все их изменения. Много приходится в корне переделывать.

Но, к счастью, мне удалось привлечь к MariaDB лучших разработчиков. С конца июля 2012, нас 18 человек, включая всех ведущих архитекторов MySQL – создатели практически всех важных функций сейчас работают на MariaDB.

LXF: Насколько это отличается от того, как вы начинали с MySQL?

МВ: Все клиенты Sun очень испугались, что же будет с их контрактами поддержки – агенты по продажам вышли из игры, заключив с важнейшими и крупнейшими пользователями MySQL соглашения сроком на 3–5 лет. А значит, года 3–4 им не нужно искать и покупать поддержку у кого-то еще.

Так что мы только используем код MySQL по GPL, и основной источник дохода – поддержка; клиентов нам не найти. По сути, мне пришлось финансировать все самому, пока срок тех контрактов не истек. Сейчас мы начали набирать клиентуру, но все это время ушло у нас на то, чтобы свести концы с концами. **LXF**



Что надо знать про патенты

Шашанк Шарма рассматривает непростую тему патентного законодательства в свете текущей саги Apple vs Samsung.

Мир людей раскололся. Они получают и предоставляют патенты на тривиальнейшие вещи якобы во имя инноваций. Время эльфов подходит к концу. Примерно так мог бы отреагировать Лорд Элронд на сложившуюся ситуацию.

Удивительный миллионный вердикт Apple vs Samsung определенно будет обжалован в Федеральном суде в Вашингтоне, но люди нескоро оправятся

от этого странного вердикта и осознают, что же нас к нему привело. Сейчас Samsung потребовал нового судебного разбирательства по ряду причин. Крайне трудно понять весь абсурдизм некоторых патентов, за последнее время полу-

«Простым смертным патенты темны, как эльфийские заклинания.»

чаемых техническими компаниями, особенно когда нам постоянно твердят о том, что смысл патентов – в продвижении инноваций.

Простым смертным патенты кажутся столь же темными, как эльфийские заклинания, и большинство из них именно так и написано, но данное руководство поможет вам разобраться в хитросплетениях патентного законодательства. Мы также обсудим различия патентных законодательств Европейского Союза и США.

Не будем вдаваться в историю и проследить появление патентов вплоть до Италии и Англии XV века, потому что не это цель нашего обсуждения, а рассмотрим, как патенты появились в Европейском Союзе. Хотя это может показаться отступлением от темы, все же важно понять, как разные нации в составе ЕС и другие страны мира рассматривают патенты.

Начнем с 1973 г., когда Западная Германия, Франция, Швейцария, страны Бенилюкс и Великобритания заключили многосторонний договор, т.н. Европейскую патентную конвенцию [European Patent Convention, EPC], сформировав единую систему патентов в Европе. Это привело к созданию Европейского бюро патентов [European Patents Office, EPO]. В Европе патент можно получить или на национальном уровне, или через EPO. Однако EPC и Европейский союз – это не одно и то же, поскольку ряд стран, входящих в состав EPC (Хорватия, Исландия, Турция и т.д.), не входят в ЕС. Заявитель может подать заявку на патент согласно преобладающему патентному законодательству отдельной страны, или EPO, если страна является членом EPC. Европейский патент [European Patent], выданный в соответствии с EPC, несмотря на свое название, имеет силу только в пределах одной страны плюс страны – члены EPC, указанные в заявке, а не на всей территории ЕС. Пока что нет положений, связанных с патентами, действительными на всей территории ЕС.

Суть патентов

Патенты подразумевают предоставление государством исключительных прав изобретателю на определенный период времени. Взамен патентообладатель дает согласие на раскрытие его изобретения. Исключительные права позволяют запрещать всем прочим использование, производство, продажу или распространение запатентованного изобретения без разрешения автора.

Точный процесс выдачи патента, срок, на который он выдается, и степень исключительности прав зависят от самого патента и от других законов данной страны, и могут быть разными в разных странах. До выдачи патента все заявки проверяются на патентоспособность, т.е. удовлетворение ряду критериев. Как минимум, изобретение должно быть новым, оригинальным и неочевидным, а его предмет – патентоспособным.

Патентное законодательство разных стран четко определяет предметы, не подлежащие па-

тентованию. Например, EPC не считает изобретениями открытия, научные теории и математические методы, и патентов на них не выдает.

В патентном законодательстве США часто используется термин «неочевидный [non-obvious]». Эквивалентный термин в европейском патентном законодательстве – «изобретательский уровень [inventive step]». Суть их в том, чтобы не предоставлять патентов на естественное и очевидное применение изобретения. Так, нельзя получить патент на применение ножовки для пиления дров. Но если кто-то применит ножовку в производстве акваланга для глубоководных исследований, это уже изобретение с высокой степенью неочевидности, или изобретение, включающее «изобретательский уровень» и имеющее право на защиту.

Другой критерий патентоспособности – «новизна». Именно здесь возникает часто применяемый термин «известный уровень техники [prior art]». Если изобретение уже было раскрыто для общества до подачи заявки на патент, это доказывает, что оно не является новым или оригинальным, и защита ему не гарантируется. Все публикации считаются разглашением, так что исследование известного уровня техники включает только свидетельство о наличии публикации об изобретении до подачи заявки на получение патента.

Патент может применяться к любому, кто использует запатентованную технологию, независимо от того, сознательное ли это нарушение или копирование запатентованной технологии. Независимая разработка устройства, нарушающего эту технологию, не считается доводом в защиту.

В Великобритании патенты выдаются на 20 лет, при условии ежегодной оплаты взноса за обновление патента. Период 18–20 лет стандартен почти для всех стран, в зависимости от начала этого периода – с даты подачи заявки или с даты выдачи патента. Требования к патентоспособности по сути одинаковы во всей Европе, и нечто непатентоспособное в Великобритании по причине объекта изобретения, скорее всего, признают непатентоспособным и в других странах Европы.

Давайте вкратце обсудим патенты на изобретение и на промышленный образец, которые стали основой юридических претензий Apple к Samsung. Все патенты, которые касаются изобретений, производящих некий полезный результат, именуется патентами на изобретение. Когда вы слышите термин «патент», обычно имеется в виду именно патент

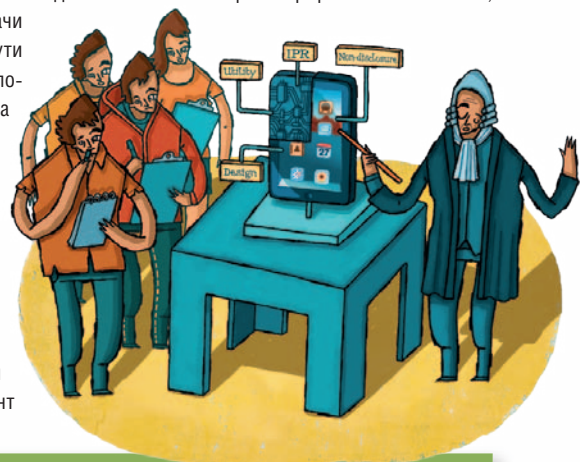
на изобретение, так как это самая общая категория инноваций. Чтобы изобретение получило патент, оно должно соответствовать одной из следующих категорий объектов изобретения:

- » **машины, содержащие движущиеся части;**
- » **произведенные объекты;**
- » **химическое соединение, такое, как химическое вещество или фармакологическое средство;**
- » **процессы, описывающие пошаговый метод (сюда относится ПО).**

Практически все изобретенное с помощью разума можно защитить патентом на изобретение, ибо единственный критерий для его получения – некий полезный результат, полученный благодаря этому изобретению. Заявка на получение патента должна содержать подробное описание, как было сделано изобретение, и сопровождаться чертежами.

Патенты на промышленный образец относятся исключительно к уникальному художественному дизайну изделия или объекта. Уникальность формы или дизайна должна присутствовать только по эстетическим причинам. Если форма несет еще и функциональную нагрузку, это уже относится к сфере патентов на изобретение. Если новая функция вводится исключительно с целью украшения и ее устранение не повлияет на работу устройства или изделия, то это патент на промышленный образец. Патенты на промышленные образцы относятся к форме или декору, увеличивающим эстетическую привлекательность объекта.

Основа претензий Apple к Samsung – патенты на промышленные образцы, выданные на iPhone и iPad. В законодательстве Великобритании нет термина «патент на промышленный образец». Его эквивалент в Великобритании – «зарегистрированный промышленный образец». Символа для обозначения зарегистрированного патента, »



Возникновение прав на интеллектуальную собственность

Впервые термин «интеллектуальная собственность» появился где-то в XIX веке, но обрел популярность только в XX веке. Термин «права на интеллектуальную собственность (IPR)» часто используется для описания множества типов законов, таких, как законы об авторских правах, патентах, торговых марках, указании происхождения, коммерческой тайне, и т.д.; все они описывают нечто, созданное с помощью интеллектуального усилия. Подобно

иным видам собственности или коммерческим активам, интеллектуальную собственность можно продать, приобрести, взять или передать в аренду. Согласно различным законам о правах интеллектуальной собственности, их владельцу гарантируются некие исключительные права для коммерческого применения на определенный период времени.

Несмотря на рост популярности прав на интеллектуальную собственность, этот термин подверга-

ется жесткой критике. Один из самых суровых критиков – Ричард Сталлмен [Richard Stallman], великий основатель СПО. Он не одобряет термина «права на интеллектуальную собственность», так как в нем свалены в одну кучу законы совершенно разного происхождения, относящиеся к разным видам деятельности. Вам все станет понятно, когда вы узнаете, что между появлением авторского права и патентов лежит период в сотню лет!



такого, как для обозначения авторского права [copyright] и торговой марки [trademark], ныне не существует, но термин «зарегистрированный промышленный образец» применяется, если форма или образец изделия зарегистрированы в Великобритании. Цель «товарного вида» в создании неповторимой индивидуальности, способствующей выделению объекта среди аналогов, т.е. вы должны, взглянув на объект, сразу узнать его, даже не глядя на ярлык. Таким образом, любое сочетание цветов, шрифтов, упаковки и маркировки, позволяющее сразу же отличить происхождение объекта, считается его товарным видом. Согласно большинству законодательств, товарный вид считается частью торговой марки, поскольку они обладают общей целью: помочь покупателю легко узнать происхождение продукта.

Патенты США и ЕС

Патенты территориальны по своей природе и подчиняются законам данной страны, однако усиливается тенденция к согласованию патентных законодательств разных стран. Соглашение ВТО по торговым аспектам прав на интеллектуальную собственность (TRIPs Agreement, ТРИПС) описывает минимальные стандарты для массы форм интеллектуальной собственности. Оно требует от стран – участниц ВТО принятия единых законов по защите авторских прав, торговых марок, патентов и т.д., и содержит положения по обеспечению процедур защиты, соблюдения и урегулирования споров. На сегодняшний день членство в ВТО требует строгого соблюдения прав интеллектуальной собственности в соответствии с Соглашением ТРИПС. Однако имеется несколько фундаментальных различий в патентном законодательстве разных стран, например, стран ЕС и США. Основное различие – «первичность заявки» против «первичности изобретения». В ЕС важнее дата подачи заявки, так как патент выдается

первому подателю заявки на его получение, даже если более поздний заявитель сделал изобретение раньше. А в США патент получает тот, кто первым сделал изобретение, независимо от даты подачи заявки. Это противоречит практически всем остальным странам. При подаче заявок на получение патента на одно и то же изобретение патент выдается тому, кто докажет, что сделал изобретение первым. Еще одно различие между законами США и ЕС касается публикации изобретения. В Великобритании, если информация об изобретении находится в публичном доступе до момента подачи заявки на получение патента, то патент не предоставляется.

Под публикацией здесь подразумеваются статьи в прессе, лекции об изобретении, предоставлении информации об изобретении инвесто-

«Усиливается тенденция к согласованию патентных законодательств.»

ру без предварительного подписания условия о неразглашении, и т.д. Для признания публикации неважно, кто изобретение обнаружил: изобретатель, один из изобретателей или независимая третья сторона. В США же изобретателям предоставляется льготный срок на год с даты публикации, за который они могут подать заявку на получение патента, а также созданы условия для получения предварительных патентов. Изобретатель может подать заявку на патент до готовности изобретения к производству, просто чтобы подтвердить более раннюю дату подачи заявки. Предварительный патент не превращается в основной автоматически. Изобретателю дается год, за который он должен подать заявку на получение соответствующего основного патента. И, наконец, в ЕС патенты на ПО не выдаются, если это не решение технической проблемы, а в США патентная защита предоставляется всем программам (см. врезку).

Зарегистрированный патент дает изобретателю право никому не разрешать производить, продавать, использовать и распространять объект патента. Суть того, что отражает патент, описывается термином «формула изобретения». Каждая

заявка на патент включает формулу изобретения, которая определяет объем патента, и в одном патенте может быть формула изобретения, состоящая из нескольких пунктов. Формула изобретения является относительно новым дополнением к патентному законодательству, особенно в странах Европы, где упоминаний о формуле изобретения в заявках на получение патента не встречалось до середины XX века. Но в США Акт о Патентах 1836 г. сделал формулу изобретения важным требованием для заявки на получение патента.

Обычная заявка на получение патента в США перечисляет все пункты формулы изобретения в конце заявки. Например, патент на изобретение Apple номер 7 469 381 имеет формулу из 20 пунктов, и оказалось, что Samsung нарушил пункт 19:

«Устройство, содержащее в себе: сенсорный экран; один или более процессоров; память; и одна или более программ, из которых одна или более программ хранятся в памяти и настроены на исполнение одним или более процессорами...»

Мы включили лишь малую часть пункта 19. Хотя формула изобретения состоит из единственного предложения, нередки формулы на полстраницы, а то и больше. Чтобы облегчить чтение, подобные формулы часто написаны в виде краткого конспекта.

Патенты на промышленный образец имеют формулу, состоящую из единственного пункта. Поскольку дизайн описан в виде чертежей в заявке на получение патента, формула изобретения обычно выглядит так: «Мы заявляем права на дизайн, отображенный на чертеже», или в подобном виде.

Вот формула изобретения на патент на промышленный образец Apple номер '889, это электронное устройство: «Мы заявляем права на художественный промышленный образец электронного устройства согласно изображению и описанию».

Ссылка на патент обычно содержит 3 последних цифры, и патент '889 – это патент USD504889.



Патенты на ПО в ЕС

Мы уже обсуждали патентоспособность и объект патента, который определяет главные области инновации, защищаемые согласно патентному законодательству. Главный критерий получения патента – новизна и полезность инновации. Согласно Статье 52 ЕПС, программы для компьютеров не считаются изобретениями и по этой причине не должны быть защищены.

Несмотря на это, Европейское патентное бюро (EPO) регулярно выдает патенты на то, что у них именуется «изобретениями, реализуемыми с помощью компьютерных технологий [computer-implemented inventions, CII]». Чтобы доказать патентоспособ-

ность такого изобретения, нужно всего лишь доказать, что оно имеет техническое происхождение и решает технические проблемы. Далее, изобретение должно быть новым, или включать элементы технического изобретения для известного уровня техники. Если перевести этот жаргон одним предложением, то это означает: если программа решает техническую проблему, ее можно запатентовать, но не в ином случае.

Лучше всего это объясняет цитата с сайта EPO: «Заявка на получение патента на систему интернет-аукционов была отклонена, поскольку система использует обычную компьютерную тех-

нологию и компьютерные сети – что означает, что она не внесла элементов технического изобретения в существующую технологию. Подобная система может способствовать коммерческому развитию своих пользователей, однако это не является тем типом развития, который требуется EPO. С другой стороны, проблема увеличения силы сигнала между мобильными телефонами – это техническая проблема, даже если она решается через модификацию ПО для телефонов, а не аппаратными средствами. Подобное изобретение будет защищено патентом, при условии, что это решение также является новым и несет в себе элементы изобретения».

КОРНИ КОНФЛИКТА APPLE VS SAMSUNG



С 2011 года Apple предъявил десятки исков Samsung и другим фирмам – изготовителям устройств, таких, как HTC, обвиняя их, среди прочего, в нарушении своих патентов и торговых марок. В свою очередь, Samsung предъявил встречный иск Apple, и с тех пор обе компании выигрывали судебные процессы в разных странах.

В своем первом иске, в США, Apple обвинила Samsung в нарушении ряда своих патентов на изобретение и на промышленный образец, и различных торговых марок: «Семейство мобильных продуктов Samsung Galaxy, появившихся в 2010 г., является примером тому. Копирование настолько явное, что продукты Samsung Galaxy на самом деле выглядят продуктами Apple – с той же прямоугольной формой со скругленными углами, серебристой окантовкой, плоской лицевой поверхностью с черными границами сверху и внизу, мягко выгнутыми краями задней части и дисплеем с цветными квадратными значками с закругленными углами. Когда телефон Samsung Galaxy используют в общественном месте, его, с очень большой долей вероятности, легко принять за продукт Apple, основываясь исключительно на внешнем виде».

Нарушение в приведенной цитате относится к различным элементам «фирменного стиля», зарегистрированным Apple в отношении iPhone:

«Регистрационное удостоверение США № 3 470 983 на общий вид продукта, включая его прямоугольную форму, закругленные углы, серебристую окантовку, черную лицевую поверхность и дисплей с 16 цветными значками».

«Регистрационное удостоверение США № 3 457 218 на конфигурацию прямоугольного портативного мобильного цифрового электронного устройства с закругленными углами».

«Регистрационное удостоверение США № 3 475 327 на прямоугольное портативное мобильное цифровое электронное устройство с серой прямоугольной частью в центре, черной полосой над и под серым прямоугольником и по изогнутым углам, и серебристой внешней окантовкой и стороной».

Изначальная жалоба, несмотря на обвинение в том, что Samsung нарушил 7 патентов на изобретение и 3 патента на промышленный образец, не включала полного списка нарушенных патентов. Например, патент '889, описывающий внешний вид iPad, был включен в форму окончательного вердикта. Присяжные вынесли решение в пользу Apple, присудив ему возмещение ущерба

в размере более \$2 миллиардов, и постановив, что Samsung сознательно нарушил ряд патентов Apple и зарегистрированный фирменный стиль.

Тот факт, что присяжные вынесли решение всего через три дня обсуждений, кажется потрясающим, особенно если учесть тот факт, что данный вердикт насчитывает 20 страниц и охватывает самые разные области законодательства – патенты, торговые марки, фирменный стиль и т.д.

На 20 страницах вердикта нашлось место для нескольких таблиц по каждому из устройств, нарушающих патенты, с более чем пятью сотнями вопросов с вариантами ответов ДА/НЕТ. Более того, окружной судья Люси Ко [Lucy Koh] предоставила присяжным более 100 страниц инструкций, подробно описывающих, чего ожидают от присяжных, что являлось свидетельством, что было доказательством, какие патенты обсуждались, и многое другое – что подчеркивает всю важность и ответственность решения, которое должно было быть принято присяжными.

От присяжных требовалось принять единодушное решение по всем вопросам, и многие юристы и консультанты признали невозможность прийти к единодушному решению по 500 вопросам всего за несколько дней, при этом законно обсуждая случаи во всей его целостности – свидетелства, аргументы и защиту. Опасаясь, что присяжные могут ошибиться в вердикте, Samsung подал ходатайство о предоставлении ему времени на изучение вынесенного присяжными вердикта, чтобы удостовериться в отсутствии в нем вопиющих ошибок. Судья сочла это ходатайство обоснованным и дала на него согласие.

Как выяснилось, присяжные и в самом деле дали маху, присудив Apple два миллиона долларов за устройство, которое, как было заключено, не нарушало патентов Apple. Присяжным предложили заново приступить к расчетам и обсуждениям тех же вопросов.

Члены жюри присяжных, в том числе и сам председатель, не раз давали интервью, периодически противореча друг другу и постоянно снабжая Samsung свидетельствами, которые он мог использовать для того, чтобы пересмотреть решение.

Например, инструкция № 35 в колоссальном перечне инструкций, насчитывающем 109 страниц, гласит: «Величина такого ущерба должна быть адекватной, чтобы возместить ущерб патентообладателя за нарушение патента. Компенсация за ущерб должна возместить патентообладателю

его финансовое состояние, в котором он находился бы, если бы не произошло такого нарушения, но в любом случае размер компенсации за ущерб не может быть ниже соразмерного роялти. Следует помнить о том, что целью компенсации за ущерб является возмещение убытков патентообладателя, а не наказание нарушителя».

Невзирая на это, председателю жюри присяжных принадлежит следующая цитата: «Мы хотели быть уверены, что наш посыл – это не просто нагоняй», и «Нам надо было убедиться, что это было достаточно болезненно, но при этом небезосновательно».

Это вопиющее нарушение инструкции судьи – лишь одна из причин, по которой Samsung требует нового суда.

Война чужими руками против Google

Тактика Apple преследования тех производителей, которые создают и продают устройства, работающие на Android, продукте Google, который, по мнению Apple, нарушает множество их патентов, именуют войной чужими руками против Google. Фактически, Стив Джобс [Steve Jobs] был твердо убежден, что Android – краденый продукт, бесовестно скопировавшим инновации Apple.

Так почему же Apple упорно преследует Samsung, вместо того, чтобы напрямую выдвинуть претензию Google? Вот несколько фактов, которые прояснят ситуацию:

Google лицензировал Android для бесплатного использования производителями, и получает доход от рекламной деятельности. Такие изготовители аппаратуры, как Samsung, извлекают доход от продажи работающих на Android устройств. В своих процессах против производителей устройств Apple заявляет об убытках вследствие появления конкурирующих продуктов. Но в этом случае вместо изначальной претензии Apple на более чем 2 миллиарда долларов, присяжные присудили компенсацию в размере миллиарда.

Еще один аспект этого дела в том, что Apple хотела получить предписания по поводу различных продуктов Samsung, которые, как заявлено, нарушают патенты Apple. Что приводит нас ко второй причине войны чужими руками.

Как только будет вынесен вердикт в пользу Apple, она может заставить Международный комитет по торговле [International Trade Commission] остановить продажу продуктов, нарушающих патенты в США. Поскольку эти продукты работают на Android, Apple сможет с успехом остановить продажу продуктов на Android в США!

Или, по крайней мере, на это Apple надеется. Поскольку Samsung подал иск на новое судебное разбирательство, нам пока что далеко до подобного результата. И, возможно, подобная ситуация не сложится в принципе. **LXF**



ZFS on Linux: Как начать применять

Алексей Федорчук продолжает рассказ о новой файловой системе – теперь в практическом аспекте.

Настоящая статья посвящена практическому использованию ZFS в Linux. Оно рассмотрено на примере openSUSE, хотя почти все из сказанного применимо и к любым другим дистрибутивам – все дистрибуцифические детали оговорены явным образом.

Обзор возможностей

Прежде чем погружаться в вопросы, связанные с ZFS, читатель, вероятно, хотел бы убедиться в том, что это стоит делать. То есть – ознакомиться с возможностями, которые будут ему предоставлены.

Для начала – немного цифр. В отличие от всех предшествовавших файловых систем и систем размещения данных, ZFS является 128-битной. То есть теоретическое ограничение на ее объем и объемы ее составляющих превышают не только реальные, но и воображаемые потребности любого пользователя. По выражению создателя ZFS, Джеффа Бонвика [Jeff Bonwick], для ее заполнения данными и их хранения потребовалось бы вскипятить океан.

Так, объем пула хранения данных (**zpool** – максимальная единица в системе ZFS) может достигать величины 3×10^{23} петабайт (а один петабайт, напомню, это 10^{15} или 2^{50} байт, в зависимости от системы счисления). Каждый пул может содержать до 2^{64} устройств (например, дисков), а всего пулов в одной системе может быть тоже не больше 2^{64} .

Пул может быть разделен на 2^{64} наборов данных (**dataset** – в этом качестве выступают, например, отдельные файловые системы), по 2^{64} каждая. Правда, ни одна из таких файловых систем не может содержать больше 2^{48} файлов. Зато размер любого файла ограничивается опять же значением в 2^{64} байт.

Количество таких ограничений можно умножить. Как уже было сказано, они лежат вне пределов человеческого воображения и возможностей. И привожу я их только для того, чтобы вселить в пользователя уверенность: ни он сам, ни его внуки и правнуки в реальности не столкнутся с ограничениями на размер файловой системы или отдельного файла, как это бывало при использовании FAT или ext2fs.

Так что перейду к особенностям ZFS, наиболее интересным, по моему мнению, десктопному пользователю. Здесь в первую очередь надо отметить гибкое управление устройствами. В пул хранения данных можно объединить произвольное (в обозначенных выше пределах) число дисков и их разделов. Устройства внутри пула могут работать в режиме расщепления данных, зеркалирования или избыточности с подсчетом контрольных сумм, подобно RAID'ам уровней 0, 1 и 5, соответственно. В пул можно включать накопители, специально предназначенные для эширования дисковых операций, что актуально при совместном использовании SSD и традиционных винчестеров.

Пул хранения становится доступным для работы сразу после его создания, без рестарта машины. В процессе работы дополнительные диски или разделы, в том числе и устройства

эширования, могут как присоединяться к пулу, так и изыматься из его состава в «горячем» режиме.

Пул хранения может быть разделен на произвольное количество иерархически организованных файловых систем. По умолчанию размер их не определяется, и растет по мере заполнения данными. Это избавляет пользователя от необходимости расчета места, потребного под системные журналы, домашние каталоги пользователей и другие трудно прогнозируемые вещи. С другой стороны, не запрещено при необходимости и квотирование объема отдельных файловых систем – например, домашних каталогов отдельных излишне жадных пользователей.

Файловые системы ZFS также доступны для размещения на них данных сразу после создания, никаких специальных действий по обеспечению их монтирования не требуется. Создание файловых систем внутри пула – процесс предельно простой: разработчики стремились сделать его не сложнее создания каталогов, и это им вполне удалось. Но при этом составляющие пула остаются именно самостоятельными файловыми системами, которые могут монтироваться со своими специфическими опциями, в зависимости от назначения.

Среди других возможностей ZFS, интересных настольному пользователю, можно упомянуть:

- » создание снапшотов файловой системы, позволяющих восстановить ее состояние в случае ошибки;
- » клонирование файловых систем;
- » компрессия данных файловой системы и дедупликация (замена повторяющихся данных ссылками на «первоисточник»);
- » создание нескольких копий блоков с критически важными данными и, напротив, возможность отключения проверки контрольных сумм для повышения скорости доступа к ним.

В общем, даже если не говорить о быстродействии ZFS (а оно весьма высоко, особенно в многодисковых конфигурациях), перечислять ее достоинства можно очень долго. Так долго, что поневоле успеваешь задаться вопросом: а есть ли у нее недостатки?

Разумеется, есть. Хотя большая их часть – скорее особенности: например, ограничения при добавлении или удалении накопителей в пуле, или отсутствие поддержки TRIM.

По большому счету, для пользователя Linux'а у ZFS обнаруживается два кардинальных недостатка: некоторая усложненность ее использования, обусловленная юридическими факторами, и высокие требования к аппаратуре.

Первый недостаток если не ликвидирован, то сглажен трудями Брайана Белендорфа [Brian Behlendorf] со товарищи и майнтейнерами прогрессивных дистрибутивов вкупе с примкнувшими к ним независимыми разработчиками. Аппаратные же претензии ZFS мы сейчас и рассмотрим.

Аппаратные потребности

Итак, ZFS предоставляет пользователю весьма много возможностей. И потому вправе предъявлять немало претензий

к аппаратной части – процессору (изобилие возможностей ZFS создает на него достаточную нагрузку), оперативной памяти и дисковой подсистеме.

Впрочем, претензии эти отнюдь не сверхъестественные. Так, процессор подойдет любой из относительно современных, начиная, скажем, с Core 2 Duo. Минимальный объем памяти определяется в 2 Гб, с оговоркой, что применение компрессии и дедупликации требуют 8 Гб и более.

Сама по себе ZFS прекрасно функционирует и на одиночном диске. Однако в полном блеске предстает при двух и более накопителях. В многодисковых конфигурациях рекомендуется разнесение накопителей на разные контроллеры: современные SSD способны полностью загрузить все каналы SATA-III, и равномерное распределение нагрузки на пару контроллеров может увеличить быстродействие.

К «железным» претензиям добавляются и притязания программные. В первую очередь, ZFS on Linux потребует 64-битной сборки этой ОС, поскольку в 32-разрядных системах действует ограничение на адресное пространство физической памяти. Кроме того, в конфигурации ядра должн быть отключена опция `CONFIG_PREEMPT`. Поэтому, например, в openSUSE ZFS может использоваться с ядром `kernel-default`, но не `kernel-desktop`, каковое, вопреки названию, устанавливается по умолчанию при стандартной настольной инсталляции.

Если вас привлекли достоинства ZFS и не уstraшили ее «железные» аппетиты, самое время опробовать ее в деле. Что потребует знакомства с некоторыми специфическими понятиями.

Терминология

» **Пул хранения данных** [zpool] – центральное понятие ZFS. В него может объединяться несколько физических устройств хранения – дисков или дисковых разделов, причем первый вариант рекомендуется. Но не запрещено и создание пула из одного диска или его раздела. В каждый пул входят

» **Виртуальные устройства** [vdev], одно или несколько. В качестве таковых могут выступать устройства без избыточности (то есть все те же диски и/или их разделы) или устройства с избыточностью – зеркала и массивы типа RAID-Z.

» **Зеркальное устройство** [mirror] – виртуальное устройство, хранящее на двух или более физических устройствах, но при четном их количестве, идентичные копии данных на случай отказа диска.

» **RAID-Z** – виртуальное устройство на нескольких устройств физических, предназначенное для хранения данных и их контроль-

ных сумм с однократным или двойным контролем четности. В первом случае теоретически требуется не менее двух, во втором – не менее трех физических устройств.

Если пул образован устройствами без избыточности (просто дисками или разделами), то одно из vdev, соответствующее ему целиком, выступает как

» **Корневое устройство**. Пул из устройств с избыточностью может содержать более одного корневого устройства – например, два зеркала.

Пулы, образованные виртуальными устройствами, служат инструментом для **наборов данных** [dataset]. Они бывают следующих видов:

» **файловая система** [filesystem] – набор данных, смонтированный в определенной точке и ведущий себя подобно любой другой файловой системе;

» **снимок** [snapshot] – моментальный снимок текущего состояния файловой системы, доступный только для чтения;

» **клон** [clone] – точная копия файловой системы в момент

его создания; создается на основе снимка, но, в отличие от того, доступен для записи;

» **том** [volume] – набор данных, эмулирующий физическое устройство, например, раздел подкачки.

Наборы данных пула должны носить уникальные имена такого вида:

```
pool_name/path/[dataset_name][@snapshot_name]
```

Пулы и наборы данных в именуются по правилам пространства имен ZFS, причем, довольно простым. Запрещенными символами для всех являются символы подчеркивания, дефиса, двоеточия, точки и процента. Имя пула при этом обязательно должно начинаться с алфавитного символа и не совпадать с одним из зарезервированных имен – log, mirror, raidz или spare (последнее обозначает имя устройства «горячего» резерва). Все остальные имена, в соответствие с демократическими традициями пространства имен ZFS, разрешены.

А вот об именах физических устройств, включаемых в пул, следует сказать особо.

Модели именования устройств

В современном Linux'е использование для накопителей имен «верхнего уровня», имеющих вид `/dev/sda`, не является обязательным, а в некоторых случаях и просто нежелательно. Однако правила менеджера устройств `udev` позволяют определять и другие модели идентификации накопителей.

В частности, штатными средствами дисковой разметки дистрибутива openSUSE предусмотрены варианты идентификации накопителей по:

» метке тома (`/dev/disk/by-label`);

» идентификатору диска (`/dev/disk/by-id`);

» пути к дисковому устройству (`/dev/disk/by-path`);

» универсальному уникальному идентификатору, Universally Unique Identifier (`/dev/disk/by-uuid`).

»

Name	Last modified
Parent Directory	
spl-0.6.0-rc11-1.1.x86_64.rpm	19-Sep-2012 04:32
spl-debuginfo-0.6.0-rc11-1.1.x86_64.rpm	19-Sep-2012 04:32
spl-debugsource-0.6.0-rc11-1.1.x86_64.rpm	19-Sep-2012 04:32
spl-modules-debugsource-0.6.0-rc11-5.1.x86_64.rpm	19-Sep-2012 04:52
spl-modules-devel-0.6.0-rc11-5.1.x86_64.rpm	19-Sep-2012 04:52
spl-modules-kmp-debug-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 04:52
spl-modules-kmp-debug-debuginfo-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 04:52
spl-modules-kmp-default-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 04:52
spl-modules-kmp-default-debuginfo-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 04:52
spl-modules-kmp-trace-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 04:52
spl-modules-kmp-trace-debuginfo-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 04:52
spl-modules-kmp-xen-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 04:52
spl-modules-kmp-xen-debuginfo-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 04:52
zfs-0.6.0-rc11-1.1.x86_64.rpm	19-Sep-2012 04:42
zfs-debuginfo-0.6.0-rc11-1.1.x86_64.rpm	19-Sep-2012 04:42
zfs-debugsource-0.6.0-rc11-1.1.x86_64.rpm	19-Sep-2012 04:42
zfs-devel-0.6.0-rc11-1.1.x86_64.rpm	19-Sep-2012 04:42
zfs-dracut-0.6.0-rc11-1.1.x86_64.rpm	19-Sep-2012 04:42
zfs-modules-debug-0.6.0-rc11-5.1.x86_64.rpm	19-Sep-2012 05:34
zfs-modules-kmp-debug-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 05:34
zfs-modules-kmp-default-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 05:34
zfs-modules-kmp-default-debuginfo-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 05:34
zfs-modules-kmp-trace-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 05:34
zfs-modules-kmp-trace-debuginfo-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 05:34
zfs-modules-kmp-xen-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 05:34
zfs-modules-kmp-xen-debuginfo-0.6.0-rc11.k3.4.6.2.10-5.1.x86_64.rpm	19-Sep-2012 05:34
zfs-test-0.6.0-rc11-1.1.x86_64.rpm	19-Sep-2012 04:42

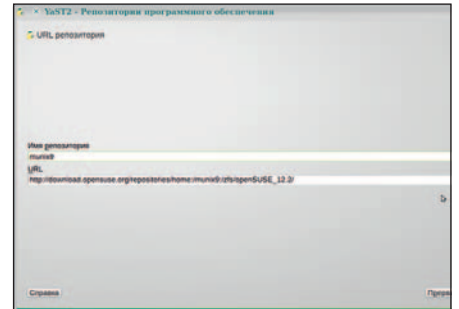
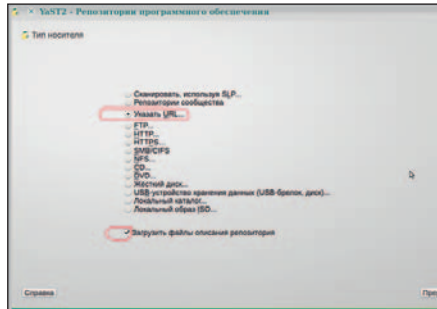
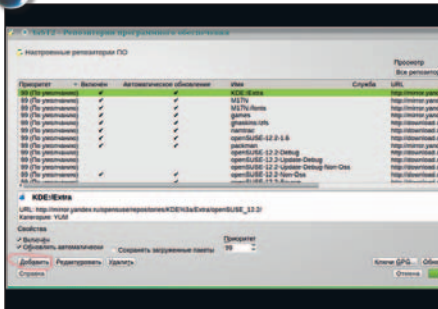
» Рис. 1. В `ghaskins`, кроме «рабочих» модулей `zfs` и `spl` для ядра `default`, можно видеть массу отладочных их сборок.

Name	Last modified
Parent Directory	
spl-0.6.0-rc11_0.40.1.x86_64.rpm	27-Sep-2012 12:11
spl-kmp-default-0.6.0-k3.4.6.2.10-rc11_0.41.2.x86_64.rpm	04-Oct-2012 06:26
spl-kmp-xen-0.6.0-k3.4.6.2.10-rc11_0.41.2.x86_64.rpm	04-Oct-2012 06:26
spl-modules-devel-0.6.0-k3.4.6.2.10-rc11_0.41.2.x86_64.rpm	04-Oct-2012 06:26
zfs-0.6.0-rc11_3.43.1.x86_64.rpm	27-Sep-2012 12:35
zfs-devel-0.6.0-rc11_3.43.1.x86_64.rpm	27-Sep-2012 12:35
zfs-dracut-0.6.0-rc11_3.43.1.x86_64.rpm	27-Sep-2012 12:35
zfs-kmp-default-0.6.0-k3.4.6.2.10-rc11_3.44.2.x86_64.rpm	04-Oct-2012 06:40
zfs-kmp-xen-0.6.0-k3.4.6.2.10-rc11_3.44.2.x86_64.rpm	04-Oct-2012 06:40
zfs-modules-devel-0.6.0-k3.4.6.2.10-rc11_3.44.2.x86_64.rpm	04-Oct-2012 06:40
zfs-test-0.6.0-rc11_3.43.1.x86_64.rpm	27-Sep-2012 12:35

» Рис. 2. Репозиторий `tunix9` существенно скромнее – имеются модули только для ядра `default` и для `xen`.



Шаг за шагом: Подключаем репозиторий через YaST



1 Заходите

Зайдите в модуль **Репозитории** центра управления YaST2 и нажмите кнопку **Добавить**.

2 Выберите источник

Перед вами откроется окно выбора источника репозитория. Не забудьте запросить файлы описания – они могут вам пригодиться.

3 Назовите имя

Введите требуемые значения в поля **Имя репозитория** и **URL**.

С полным списком вариантов идентификации блочных устройств можно ознакомиться, просмотрев имена подкаталогов в каталоге `/dev/disk/`; их содержимое – это символические ссылки на имена «верхнего уровня».

С идентификацией по метке тома и по UUID, вероятно, знакомо большинство читателей. И к тому же в пространстве имен ZFS они не используются. А вот с идентификацией *by-path* и *by-id* нужно познакомиться поближе.

Модель именования *by-path* использует имена устройств, привязанные к их положению на шине PCI и включающие номер шины и канала на ней. Имя дискового устройства выглядит приблизительно так:

```
pci-0000:00:1f.2-scsi-0:0:0
```

Дисковые разделы маркируются добавлением к имени устройства суффикса *part#*.

Модель именования *by-path* идентифицирует устройства вполне однозначно, и особенно эффективна при наличии более чем одного дискового контроллера. Однако сами имена и устройств, и разделов описываются довольно сложной для восприятия последовательностью. Да и в большинстве «desktopных» ситуаций модель эта избыточна.

Модель идентификации *by-id* представляет имена носителей информации в форме, наиболее доступной для человеческого понимания. Они образованы из названия интерфейса, имени производителя, номера модели, серийного номера устройства и, при необходимости, номера раздела, например:

```
ata-SanDisk_SDSSDX120GG25_120823400863-part1
```

Таким образом, все компоненты имени устройства в модели *by-id* определяются не условиями его подключения или какими-то правилами, а задаются производителем и жестко прошиты в «железе». И потому эта модель является наиболее однозначной

для именования устройств. А также, что немаловажно, строится по понятной человеку логике. Не случайно именно она принята по умолчанию в инсталляторе openSUSE.

Какую из моделей именования устройств выбрать для данного пула – зависит от его назначения и масштабов. Имена «верхнего уровня» целесообразно применять для однодисковых пулов (особенно если в машине второго диска нет и не предвидится, как обычно бывает в ноутбуках). Они же, по причине удобопонятности и простоты, рекомендуются для экспериментальных и разрабатываемых пулов. И очень не рекомендуются – во всех остальных случаях, так как зависят от условий подключения накопителей.

Этого недостатка лишена модель *by-id*: как пишет Брайан, при ее использовании «диски можно отключить, случайно смешать и подключить опять произвольным образом – и пул будет по-прежнему корректно работать». Как ее недостаток рассматривается сложность конфигурирования больших пулов с избыточностью. И потому она рекомендуется для применения в «desktopных» и «квартирных» (типа семейного сервера) условиях.

Для больших (более 10 устройств) пулов из дисков, подключенных к нескольким контроллерам, рекомендуется идентификация *by-path*. Однако в наших целях она громоздка и избыточна.

Наконец, ZFS on Linux предлагает и собственную модель идентификации – `/dev/disk/zpool`, в котором именам *by-path* ставятся в соответствие уникальные и осмысленные «человекочитаемые» имена, даваемые пользователем. Модель эта рекомендуется для очень больших пулов, каковых на настольной машине ожидать трудно.

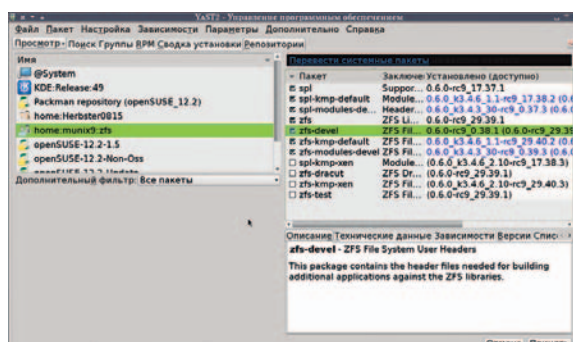
Так что дальше я буду использовать имена «верхнего уровня», говоря об абстрактных или экспериментальных ситуациях, и об именах *by-id*, когда речь пойдет о практических примерах применения ZFS.

Включение поддержки ZFS

Для практического использования ZFS on Linux перво-наперво необходимо обеспечить ее поддержку в вашем дистрибутиве – ибо по причинам, описанным в предыдущей статье, сама собой она не поддерживается ни в одном Linux'е.

Как это сделать, зависит от дистрибутива. В Сети можно найти подробные инструкции для Ubuntu и Gentoo, которые легко распространяются на клоны обеих систем. Не столько инструкции, сколько руководства к самостоятельному действию имеются на сайте проекта ZFS on Linux для абстрактных RPM- и Deb-based дистрибутивов. Я же расскажу о том, как это делается в openSUSE релизов 12.1 и 12.2.

Рис. 3. Пример дан для *minix9*, но из *ghaskins* требуются те же компоненты.



Как вы наверняка догадались, ZFS не поддерживается в openSUSE ни «искарюпки», ни в официальных репозиториях. Но зато в репозиториях неофициальных, так называемых «домашних», пакеты ее поддержки представлены аж в двух экземплярах: в *munix9* и в *ghaskins*. Точные их адреса легко найти через систему OBS (Open Building System) по ключевому слову *zfs*.

Какому из репозиториях отдать предпочтение – вопрос спорный. Первые свои опыты с ZFS on Linux я проводил, основываясь на пакетах из *munix9*. И они прошли без всяких осложнений, хотя и велись в сугубо экспериментальном режиме. К моменту понимания, что эта система для меня – «всерьез и надолго», последняя тогда версия *zfs* имела только в репозитории *ghaskins*. Однако его использование требует некоторых дополнительных манипуляций.

Кроме того, в репозитории *ghaskins* на данный момент имеются пакеты только для openSUSE релизов 12.1 и 12.2. Репозиторий же *munix9* охватывает все актуальные ныне версии SLE и openSUSE, включая Tumbleweed и Factory.

Различаются репозитории и набором пакетов (рис. 1 и 2). Так что окончательный выбор я предоставляю читателю. Но на какой бы репозиторий он ни пал, его следует подключить. Тех, кто с этим затрудняется, отсылаю ко врезке.

Подключение репозитория

Сделать это можно любым из трех способов. Первый – через командную строку, с помощью *zypper*'а:

```
# zypper ar -f [URL] [Name]
```

Второй – через центр управления YaST2 (см врезку «Шаг за шагом» на предыдущей странице).

После подключения репозитория надо будет установить (с помощью *zypper*'а или модуля управления пакетами *YaST*'а) дополнительные компоненты (рис. 3).

Наконец, третий способ, для самых ленивых – отыскать пакеты *zfs*, *spl* и сопутствующие через OBS и прибегнуть к «установке в один клик». В этом случае подключение репозитория будет совмещено с установкой пакетов.

Возможно, не вредным окажется и пакет *zfs-test*. А вот *zfs-dracut*, предназначенный для создания *initrd* с поддержкой ZFS, несмотря на его потенциальную нужность, установить не удастся: требуемый для него пакет *dracut* в openSUSE пока не поддерживается.

Следует учесть, что при использовании ядра *kernel-desktop* (а скорее всего, так оно и есть) пакет *zfs-kmp-default* потянет за собой и соответствующее ядро *kernel-default*. Пункт загрузки которого будет внесен в меню *Grub*, но не будет отмечен как умолчательный – этим надо озаботиться самому.

И, наконец, при использовании пакетов из *ghaskins* потребуются, скорее всего, сделать в каталогах */etc/init.d/rc3.d* и */etc/init.d/rc5.d* символические ссылки на файл */etc/init.d/zfs*. Иначе файловые системы ZFS, к созданию которых мы приближаемся, не будут автоматически монтироваться при старте и размонтироваться при останове системы.

При использовании репозитория *munix9* эти действия будут выполнены в ходе установки пакетов нечувствительно для пользователя.

Вот теперь можно приступать к применению ZFS в мирных практических целях. Что мы с вами и предпримем – но уже в следующем номере. **ixp**

Подпишись на журнал Linux Format

**LINUX
FORMAT**
Главное в мире Linux

Оформи подписку на 2013 год на печатную версию журнала Linux Format и получи бонусы!

Бонусы каждому редакционному подписчику

DVD с архивом номеров 2005–2012

Подписка на PDF-версию журнала

Объемная наклейка на системный блок



Стоимость подписки

- » Годовая — 2280 руб.
- » Полугодовая — 1230 руб.

Цены указаны без учета стоимости доставки

Варианты доставки

- » Почтой по России (от 38 руб. за один экземпляр)
- » Курьером по Москве (350 руб.) и Санкт-Петербургу (250 руб.)

- » Экспресс-доставка курьерской службой «СПСР» по России
- » Самовывоз из офисов в Москве или Санкт-Петербурге

Адреса и телефоны редакции

Санкт-Петербург, Лиговский пр., 50, корп. 15
Тел.: (812) 309-06-86

Москва, Красноказарменная ул., 17
Тел.: (499) 271-49-54

shop.linuxformat.ru



Что за штука...

Проект Byzantium

Маянк Шарма представляет проект, который поможет вам оставаться на связи в пост-апокалиптическом мире, где кишат зомби.

В Зомби... вы это серьезно? Или решили пошутить?

О Проект Byzantium предназначен для поддержания взаимосвязи с миром, если официальные каналы вдруг станут недоступны из-за природного катаклизма, действий репрессивного правительства... или зомби.

В Но почему выбрано такое название — Byzantium?

О Название дано в честь известного византийского инженерного механизма обеспечения отказоустойчивости.

В Что? А нельзя ли объяснить как-нибудь попроще?

О Отказоустойчивость, в инженерии, означает, что система продолжает функционировать, даже если некоторые ее части выходят из строя. «Задача византийских генералов, один из которых предатель» — интересный сценарий такого рода.

В И какая связь между отказоустойчивостью и этим проектом?

О Разработчики проекта Byzantium надеются создать сеть, которая останется работоспособной, даже если некоторые ее составные части

выйдут из строя. То есть при потере одного из узлов контроль просто переходит к остальным, которые его компенсируют, а значит, вы остаетесь на связи.

В Здорово! А как они планируют создать такую сеть?

О Разработчики Project Byzantium утверждают, что они не изобретают новые технологии, предпочитая комбинировать старые неожиданным, ранее не встречавшимся способом. Сеть Byzantium будет беспроводной распределенной динамической сетью.

В Так... и что этот трескучий набор слов означает?

О ОК, давайте я вам поясню. Динамическая сеть – это беспроводная сеть децентрализованного типа, а значит, она не зависит от роутеров или точек доступа. Вместо этого информация передается от одного узла к другому, и так далее.

В А что такое, в свою очередь, распределенная сеть?

О Считайте это расширенным вариантом динамической сети. В распределенной сети ряд

пуская узел, чтобы подключиться к сети. Преимущество динамического соединения в том, что в нем невозможна цензура, поскольку оно не использует инфраструктуру остального Интернета.

В Как же мне запустить собственный узел Byzantium?

О Проект Byzantium выпускает live CD на основе облегченного дистрибутива Porteus Linux, двоично-совместимого со Slackware. Сделать из своего компьютера узел Byzantium можно, загрузив образ через CD или USB.

только сервис выведен в сеть, он появляется на заглавной странице узла. Затем информация передается другим узлам, которые автоматически обновят свои заглавные страницы, добавив туда ссылку на новый сервис.

В Раз это live CD, можно ли будет сохранить свою конфигурацию?

О Да, можно. В дистрибутиве Porteus уже имеется утилита для создания зашифрованного хранилища на съемном носителе. Планируется, что оно будет доступно с панели управления, чтобы при последующих запусках пользователи могли продолжить с того же места, на котором закончили.

В Что ж, тонко. А клиентам нужно будет что-то делать, чтобы подключиться к узлу?

О Совсем нет. Со стороны клиента достаточно иметь устройство, поддерживающее подключение к динамической сети. По сути, любой ноутбук это умеет, а вот некоторые смартфоны потребуют установки дополнительного ПО. Чтобы упростить задачу, вы можете подцепить свой узел Byzantium к обычному беспроводному маршрутизатору, чтобы клиенты могли подключаться к нему, как к любой точке доступа Wi-Fi.

В Помимо сервисов в составе сети, могу ли я выходить в Интернет?

О Конечно, можете! В том-то и прелесть смешанных протоколов, применяющихся в Byzantium. Если узел находит свободное подключение, он может выступать в качестве шлюза, предоставляя другим клиентам доступ в Интернет. На данный момент, идет работа по добавлению этой функции в грядущие релизы.

В И на какой стадии сейчас находится этот проект?

О Текущий релиз Byzantium Linux – 0.2a, и скачать его можно с сайта проекта: <http://project-byzantium.org>. Хотя дистрибутив пока на стадии альфа, он полностью пригоден для создания смешанной сети, но разработчики на данный момент не советуют применять его в реальных экстренных ситуациях. Помимо web-приложения *status.net*, он также включает *Mplayer* и *Firefox*, поверх облегченного рабочего стола *LXDE* или полноценного KDE 3.5. Вы можете установить его на USB-накопитель или на жесткий диск при помощи Porteus, в котором пока что отсутствует менеджер разделов. Сетевая панель конфигураций позволяет настраивать смешанную сеть а также включать и выключать различные службы. **LXF**

«Предустановлены будут приложения, популярные у знатоков всего мира.»

узлов предназначен для маршрутизации трафика, помимо того, что они же являются клиентами сети. Это противоречит традиционной сети, где есть шлюз по умолчанию. Если этот шлюз недоступен, сеть оказывается изолирована. А в распределенной сети такого шлюза нет, и каждый узел потенциально выполняет эту функцию по отношению к другим.

В Поправьте меня, если я ошибаюсь, но это выглядит неэффективным, особенно для большой сети, где все узлы постоянно пытаются обновить статус своих пиров.

О Да. Протоколы, использовавшиеся в ранних версиях, были малоэффективны. Но с годами эти ограничения были устранены, что позволило создавать более эффективные сети. Сейчас существует уже свыше 70 протоколов для распределенных сетей!

В Довольно много. И который используется в Byzantium?

О На данный момент, разработчики используют два. Это протокол Babel, очень эффективный для поддержания участников соединения в актуальном состоянии относительно текущей конфигурации сети. Еще один – BATMAN, или Better Approach To Mobile Adhoc Networking, децентрализирующий осведомленность о маршрутизации внутри сети.

В Давайте проверим, правильно ли я понял. Сеть Byzantium состоит из узлов и клиентов, объединенных на динамической основе?

О Именно так. Кроме того, узлы могут выступать и в роли клиентов, и вам не нужно за-

В А какое оборудование нужно для этого византийского Linux?

О Он запустится на любом компьютере с архитектурой x86, но для работы с web-приложениями понадобятся соответствующие ресурсы. Разработчики рекомендуют для обычного узла Byzantium иметь процессор не ниже i586 и не менее 1 Гб ОЗУ. И, очевидно, потребуется как минимум одна карта Wi-Fi.

В А моя собственная карта будет поддерживаться?

О Изначально дистрибутив включает ядро Linux версии 3.1.8 и драйверы для различных беспроводных карт. Есть вероятность, что если карта поддерживается вашим текущим дистрибутивом, то будет работать и в Byzantium Linux.

В То есть в Byzantium будут поддерживаться все web-сервисы?

О Проект Byzantium находится на ранней стадии, и с окончательным набором приложений пока не определились. Но предустановлены и настроены будут приложения, самые популярные у знатоков всего мира. Для микроблоггинга решено использовать *status.net*; будут также приложения для создания социальной сети, Wiki и блога. Планируются еще клиент мгновенных сообщений и текстовый процессор для коллективной работы.

В Каким образом можно управлять этими сервисами?

О В Byzantium Linux есть сетевая панель-управления, через которую администраторы узла смогут активировать и настраивать сервисы. Как



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.

Расширение учебного портфолио

Linux Foundation предлагает несколько учебных программ, но если раньше они предназначались для настоящих экспертов в области ядра (например, «Внутреннее устройство и отладка ядра Linux» или «Отладка драйверов устройств в Linux»), то недавно учебное портфолио было расширено и включило курсы для пользователей и администраторов, такие как «Администрирование Linux» или «Архитектура облака и его развертывание». Компания предлагает и более короткие программы продолжительностью от двух часов до двух дней, касающиеся щекотливой темы совместимости открытого ПО, в которой я так несведущ, что даже не знаю, что с чем должно быть совместимо или (ближе к теме) почему совместимость открытых программ обеспечить сложнее, чем совместимость закрытых.

Предлагаются также курсы по разработке для Android, и Доминик Дюваль [Dominic Duval] (глава отдела по обучению для корпоративных клиентов Linux Foundation) обещает, что в новом году откроются дополнительные.

Хотя компания в основном фокусируется на местном рынке обучения для корпоративных клиентов (в противовес «публичным мероприятиям»), она проводит открытые курсы в Барселоне за день до LinuxConf (которая к моменту чтения этой статьи могла уже пройти).

Сменим тему. Недавно я поменял свой iPhone на телефон с Android, и теперь могу подключить его к ноутбуку и скопировать на него музыку через командную строку. Аллилуйя! Почему я так долго терпел iTunes, остается загадкой.

Может, в Apple и делают прекрасные интерфейсы, но их приковывание к себе пользователя и склонность к судебным искам с большими деньгами мне несимпатичны. В мыслях у меня некий жест с двумя пальцами, но боюсь его показывать. Вдруг у Apple есть на него патент.

chris.linuxformat@gmail.com

Эзотерическое системное администрирование из причудливых заворотов кишок серверной



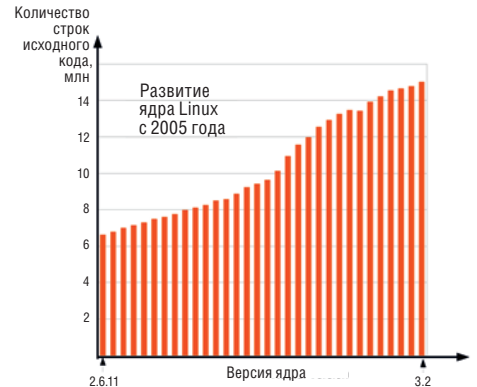
Подсчет ядер

За кулисами энергоблока сообщества, управляющего разработкой ядра Linux.

Вы слышали о Марке Брауне [Mark Brown] или Томасе Гляйкснере [Thomas Gleixner]? Думаю, нет. Как и я. Но согласно последнему отчету от Linux Foundation, эти два разработчика внесли наибольшее количество изменений в ядро, начиная с версии 2.6.5.

Исходные данные для отчета формируются путем мониторинга изменений на git.kernel.org. Исследуются вопросы «Быстро ли это происходит?», «Кто это делает?», «Что они делают?» и «Кто спонсор?». В ядре Linux более 15 миллионов строк кода – это самый крупный проект совместной разработки в истории вычислительной техники. Для сравнения, во всех семи книгах о Гарри Поттере – чуть больше миллиона слов.

Многие студенты моих курсов спрашивают: «Кто управляет ядром?» Бездумный ответ – «Никто», но на деле существует хорошо отлаженный процесс рассмотрения и утверждения патчей-заплаток ядра (кстати, «patch» – официальный термин для описания набора изменений: например, для добавления новой функции, под-



» За последние семь лет объем ядра Linux вырос с 6,6 до 15 миллионов строк кода.

держки нового устройства, исправления ошибки или увеличения производительности). Обычно заплатки рассматриваются всей командой, затем утверждаются разработчиками, отвечающими за соответствующую подсистему, и, наконец, принимаются в основной дистрибутив Линусом Торвальдсом – у него исключительное право решать, что войдет в следующий релиз ядра.

Немного цифр:

- » 226 – столько компаний работает над ядром 3.2.
- » 37626 – количество файлов в ядре 3.2.
- » 14 – средний период времени в минутах между применением патчей к ядру.
- » 78 – среднее число дней от выпуска до выпуска.
- » 12243 – максимальное количество патчей, которое когда-либо применялось к версии ядра.
- » 17,9 – изменения, приходящиеся на долю индивидуальных разработчиков (%).

Как узнать больше

Отчет доступен на сайте <http://bit.ly/Ha4nRJ>. Подробное описание процесса разработки ядра см. по ссылке <http://bit.ly/Xt0c8N>. Если вы хотите принять участие в разработке, у этих ребят есть по-настоящему серьезные и трудные обучающие программы.

Итак, вы хотите стать сисадмином?

Пятая часть серии, которая превратит вас из новичка в звезду системного администрирования. Сейчас мы установим web-сервер.

На конец в этой серии мы подошли к тому, что должны делать серверы – обслуживать! Я решил установить web-сервер (хотя и не обычный *Apache*), но попытался обобщить процесс так, чтобы эти навыки пригодились вам при установке любого сервиса.

На всех уроках этой серии мы пользуемся CentOS 6.2. Если вы хотите следовать за мной, установите CentOS (можно и в виртуальную машину) в соответствии с описанием из первой статьи. У многих команд из обсуждаемых в этом месяце большой объем выходных данных, и на все здесь не хватит места. Если вам нужны подробности, то полный транскрипт многих команд есть на нашем сайте: www.linuxformat.com/archives?issue=165, так что смелее обращайтесь к нему в соответствующих местах.

Общая картина

Конечно, детали установки и настройки сервиса сильно зависят от разновидности сервиса, но в целом для этого нужно выполнить следующие шаги:

- » Установить сервис.
- » Найти и прочесть его man-страницу (иногда для файла настройки есть отдельная man-страница).
- » Изменить его конфигурацию в соответствии со своими потребностями.
- » Настроить запуск сервиса при загрузке системы.
- » Запустить сервис вручную.
- » Проверить наличие сообщений об ошибках и/или об успешном запуске в лог-файле.
- » Проверить сервис.

Начинаем

В моей CentOS 6 был по умолчанию установлен классический HTTP-сервер *Apache*, но я решил воспользоваться другим сервером – *Lighttpd*. На его сайте (lighttpd.net) написано следующее: «*Lighttpd* – надежный, быстрый, послушный и очень гибкий web-сервер, оптимизированный для высокопроизводительного окружения. У него очень небольшие требования к памяти по сравнению с другими web-серверами, и он бережно расходует ресурсы процессора». Озвучивают его имя как “lighty [легкий]”.

Так как *Apache* уже установлен на моем компьютере, нужно убедиться, что он не запущен, и не запускать его во время загрузки. Если запустить и *Lighttpd*, и *Apache*, они оба попробуют слушать порт 80, а это недопустимо.

```
# service httpd status
httpd is stopped
# chkconfig httpd --list
httpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

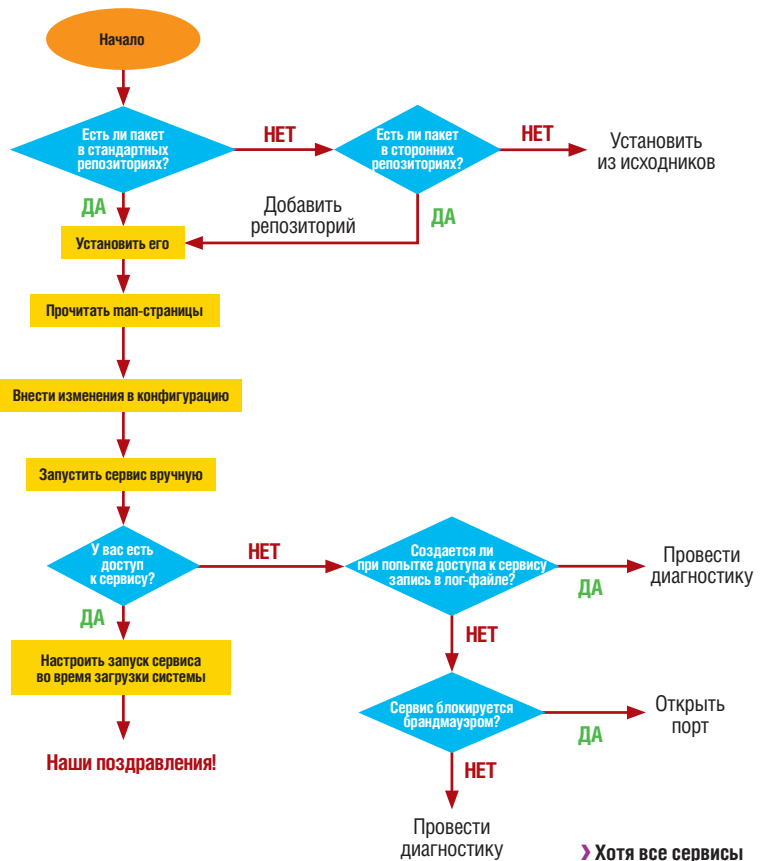
Сервис отключен – отлично. Но чтобы быть полностью уверенными, лучше удалить **httpd** совсем:

```
# rpm --erase httpd gnome-user-share
```

Почему я также удалил пакет *gnome-user-share*? Потому что **rpm** говорит, что он зависит от **httpd**, и я был вполне уверен, что он мне не нужен.

Надежно избавившись от *Apache*, перейдем к установке *Lighttpd*. Оказывается, что последнего нет в официальных репозиториях CentOS, как показывает быстрый поиск:

```
# yum search lighttpd
No Matches found
```



» Хотя все сервисы разные, их установка обычно состоит из таких этапов.

Но, введя в Google “CentOS6 lighttpd”, я нашел *rpmforge* и загрузил оттуда небольшой **rpm**:

```
# wget http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el6.rf.i686.rpm
```

Это версия для 32-битных систем. Чтобы загрузить 64-битную версию, измените *i686* на *x86_64*. Команда **rpm -qlp** с этим пакетом показала, что это не пакет *lighttpd*, он просто содержит файл **.repo**, указывающий на репозиторий *rpmforge*. Он также содержит файлы определения репозитория для APT, утилиты управления пакетами Debian, но здесь они нам не нужны.

```
Установим пакет:
# rpm -i rpmforge-release-0.5.2-2.el6.rf.i686.rpm
```

Устранение ошибок

Конечно, при первом запуске ваш сервис будет работать отлично, и вы можете пойти погулять и полюбоваться барашками в небе. Но если что-то пойдет не так, вот несколько стандартных вещей, которые стоит проверить:

- » Запущен ли сервис на самом деле? (Поищите его командой **grep** в выводе **ps -ef**.)

- » Открыт ли нужный(е) порт(ы)? (На этот вопрос ответит **lsof -i**.)
- » Открыт ли порт в брандмауэре? (Запустите **iptables -L** или воспользуйтесь системной утилитой настройки брандмауэра *systemconfig-firewall*.)
- » Появляются ли какие-то сообщения в лог-файле с момента запуска сервиса до попытки обращения к нему?

Получение справки

Если вы пришли в Linux из мира Windows, придется кое-что изменить в голове, так как вы оказались в экосистеме справки, окружающей открытое ПО. Трудное найти открытую программу, у которой не было бы активного сообщества, с радостью готового помочь, если спросить правильно.

Часто есть вики или форум, на котором можно задать вопросы (чтобы открыть форум *Lighttpd*, просто щелкните по ссылке на домашней странице). У многих программ также есть каналы чата (в основном на irc.freenode.net), хотя, по моему опыту, отношение сигнал/шум в них очень низкое.

```
warning: rpmforge-release-0.5.2-2.el6.rf.i686.rpm: Header V3 DSA/SHA1 Signature, key ID 6b8d79e6: NOKEY
```

Вы увидите, что *rpm* жалуется на невозможность проверить цифровую подпись пакета, так как у нее нет соответствующего публичного ключа (на самом деле, ключи есть в пакете!). Теперь поиск с *yum* более удачен:

```
# yum search lighttpd
```

```
lighttpd.i686 : Lightning fast webserver with light system requirements [Молниеносный web-сервер со скромными требованиями к системе]
```

и мы можем установить его:

```
# yum install lighttpd
```

Так, пока неплохо. Посмотрим, что у нас есть, перечислив файлы в пакете:

```
# rpm -ql lighttpd
```

```
... здесь появится длинный список файлов ...
```

Этот список дает ответы на несколько важных вопросов (не забудьте, что полный транскрипт есть на нашем сайте):

» Какие исполняемые файлы здесь есть? Их два: **lighttpd** и **lighttpd-angel** (что бы это ни было).

» Где находятся файлы настройки? Файл настройки верхнего уровня – **/etc/lighttpd/lighttpd.conf**, а файлы настройки для отдельных модулей находятся в каталоге **/etc/lighttpd/conf.d**.

» Куда помещаются файлы, которые должен обслуживать сервер? В каталог **/srv/www/lighttpd** (согласно официальному стандарту иерархии файловой системы – Filesystem Hierarchy Standard, им вообще-то должен быть **/srv/www**, но многие системы не следуют этому правилу).

» Есть ли *map*-страницы? Да, есть одна страница под названием **lighttpd**.

» Это птичка? Самолетик? Нет, это *Lighttpd*, «легкая» альтернатива *Apache*.



LIGHTTPD

fly light.

» Есть ли лог-файл? Да, это **/var/log/lighttpd** (на самом деле оказалось, что это целый каталог с лог-файлами).

» Есть ли другая документация? Да, в каталоге **/usr/share/doc/lighttpd-1.4.28** есть много текстовых файлов.

Следующий шаг – ознакомиться с *map*-страницей. Она довольно короткая, так как лишь описывает параметры командной строки и отправляет нас к файлам **/usr/share/doc** за более подробной информацией. Это начинает казаться долгой каторгой, а я хочу немедленного удовлетворения. Поэтому для минимальной демонстрации работы сервера я сделал вот что.

Во-первых, добавил одну строку текста в файл **/srv/www/lighttpd/index.html**, чтобы у сервера было что обслуживать. При желании можете добавить туда много хитроумного HTML-кода, но для этого теста мне было достаточно строки:

```
This is a test from Chris! [Это тест Криса!]
```

Затем попытался запустить сервер:

```
# service lighttpd start
```

```
Starting lighttpd: 2012-09-06 15:16:43: (server.c.722) couldn't set 'max filedescriptors' Permission denied
```

Ай-яй-яй. Что здесь не так? Сообщение об отсутствии прав доступа от процесса, запущенного от имени суперпользователя-*root*, необычно и скорее всего означает, что операцию запрещает SELinux. Чтобы это проверить, я попробовал отключить SELinux и снова запустить сервер:

```
# setenforce 0
```

```
# service lighttpd start
```

```
Starting lighttpd: [ OK ]
```

Теперь все хорошо. Если вы подумали: «Какое умное и пронизательное предположение!», скажу, что за последние несколько лет работы с системами в стиле RedHat 6 (в которых SELinux по умолчанию работает в принудительном режиме Enforcing) каждый раз, когда что-то не работает и я не могу найти веской причины, моя обычная реакция – попробовать отключить SELinux. Это не совсем правильное решение, но в нашем случае оно сработало.

Хороший способ узнать, запущен ли сервер – проверить, слушает ли он заданный порт (в данном случае, 80). Это легко:

```
# lsof -i TCP:80
```

```
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
```

```
lighttpd 2030 lighttpd 4u IPv4 15277 0t0 TCP *:http (LISTEN)
```

– у нас все в порядке.

Учтите, запуск сервиса командой **service** не означает, что он будет запускаться при загрузке системы. Для этого нужно выполнить еще одну команду:

```
# chkconfig lighttpd on
```

Открываем порт

Прежде чем подключиться к серверу, нужно сделать еще одну вещь – открыть порт 80 в брандмауэре. Это легко сделать утилитой *system-config-firewall* (см. экранный снимок на стр. 61), которой можно воспользоваться, даже если у вас есть только *ssh*-доступ к серверу. (Не закройте случайно порт для *ssh*!). Если теперь открыть в браузере адрес сервера (<http://192.168.1.64> – у вас, конечно, IP-адрес будет другим), на экране должен появиться текст, который был добавлен в **index.html**.

Итак, вот оно – наше немедленное удовлетворение! Мы смогли сделать это, не заглядывая в файлы настройки; но у вас вряд ли получится продвинуться намного дальше, как следует не разобравшись в них и не изменив их. Файл настройки верхнего уровня *lighttpd* (**/etc/lighttpd/lighttpd.conf**) – типичный пример такого файла, полный пояснений-комментариев и закоментированных записей. Иногда из-за этих комментариев сложно увидеть записи, поэтому вот маленький прием, который поможет от них избавиться:

```
# grep -v '^#' /etc/lighttpd/lighttpd.conf | grep -v '^$'
```

Эта команда исключит из содержимого файла комментарии и пустые строки. Выберем несколько основных строк:

```
var.server_root = "/srv/www"
server.port = 80
server.username = "lighttpd"
server.document-root = server_root + "/lighttpd"
```

Они задают соответственно:

» Базовый каталог. Несколько других важных каталогов задаются относительно него.

» Номер порта, который будет слушать сервер.

» Пользователь, от имени которого будет запускаться сервер. Изначально он запускается от имени суперпользователя-root (чтобы связаться с портом 80); затем переключается на указанного пользователя. Этот пользователь добавляется в `/etc/passwd` при установке пакета.

» Каталог, содержащий обслуживаемое содержимое.

В файле `lighttpd` также часто задаются параметры производительности. Пример из этого файла – строка

```
server.max-connections = 1024
```

Похоже, что разработчики задали их значения правильно, и я не советовал бы вам менять их, если только не (а) вы знаете, что делаете, и (б) вы сможете объективно измерить любые изменения в производительности при изменении параметров.

Таинственный сад

Конечно, теперь можно сделать многое другое – включить скрипты PHP CGI, настроить несколько виртуальных хостов и т.д. Каждое из этих действий снова погрузит вас в файлы. На самом деле при установке любого сервиса вы попадаете в новый мини-мир синтаксиса файлов настройки. Это как открыть дверь в таинственный сад, где вы доселе не бывали.

Прежде чем закончить, вспомним о SELinux. Как вы помните, мы отключили его, чтобы быстро все поправить. Однако, как бы я ни относился к SELinux, он свое дело делает, и просто отключить его – не лучшая идея. Вместо этого нужно разработать новую политику SELinux, которая позволила бы `Lighttpd` делать то, что ему необходимо. Оказывается, это относительно просто. Вот что нужно сделать: сначала с помощью `auditd` мы запишем в журнал те действия, которые `lighttpd` пытается выполнить при запуске. Затем воспользуемся маленькой удобной программой `audit2allow` (из пакета `policycoreutils-python`), чтобы преобразовать запрещенные операции, записанные `auditd` в лог-файл, в правила политики SELinux. Наконец, мы установим новые правила с помощью `semodule`.

Команды приводятся ниже. Не забудьте заглянуть в полный транскрипт за подробностями.

Сначала переформируем всю политику SELinux. Это может занять минуту или две:

```
# semodule -DB
```

Мне также пришлось установить программу `audit2allow`:

```
# yum install policycoreutils-python
```

Эта команда установила еще восемь пакетов для разрешения зависимостей. Затем перезапустим демон `auditd`:

```
# service auditd restart
```

Теперь снова включим SELinux и попробуем перезапустить `lighttpd`. Да, снова ничего не получится, но нам нужно поймать запрещающие сообщения. Они будут записаны в `/var/log/audit/audit.log`:

```
# setenforce 1
```

```
# service lighttpd restart
```

Вот хитрый момент. Мы извлекаем соответствующие сообщения, записанные `auditd`, и передаем их `audit2allow`, которая преобразует их в правила политик SELinux:

```
# grep lighttpd /var/log/audit/audit.log | audit2allow -M lighttpdmaxfds
```

Чтобы узнать больше

Почему бы вам не установить собственный сервис, следуя общим принципам, которые мы изучили? Я бы посоветовал `Vsftpd` или `Proftpd`. Оба имеются в репозиториях CentOS. Если у вас есть старые номера

LXF или возможность загрузить электронные версии по подписке, см. в **LXF132** мое руководство по настройке `Vsftpd`. Если хотите задачу посложнее, попробуйте установить `bind` (сервер DNS) или `MySQL`.

Сейчас в текущем каталоге появился файл `lighttpdmaxfds.te`. Это обычный текстовый файл. Также появится скомпилированная (двоичная) версия в `lighttpdmaxfds.pp`. Ее-то мы и загрузим:

```
# semodule -i lighttpdmaxfds.pp
```

Теперь `lighttpd` нормально запускается с включенным SELinux:

```
# service lighttpd restart
```

```
Stopping lighttpd: [FAILED]
```

```
Starting lighttpd: [ OK ]
```

Наконец, принудительно переформируем политику SELinux:

```
# semodule -B
```

(благодарю за помощь Google и некоего FLO на сайте Warp1337).

У меня осталось место только на пару слов о лог-файлах, о которых за всем этим мы позабыли. Для таких сервисов, как `lighttpd`, которые пишут в свои журналы напрямую (а не через `rsyslog`), лог-файлы легко найти, выведя список файлов, открытых web-сервером, например, так:

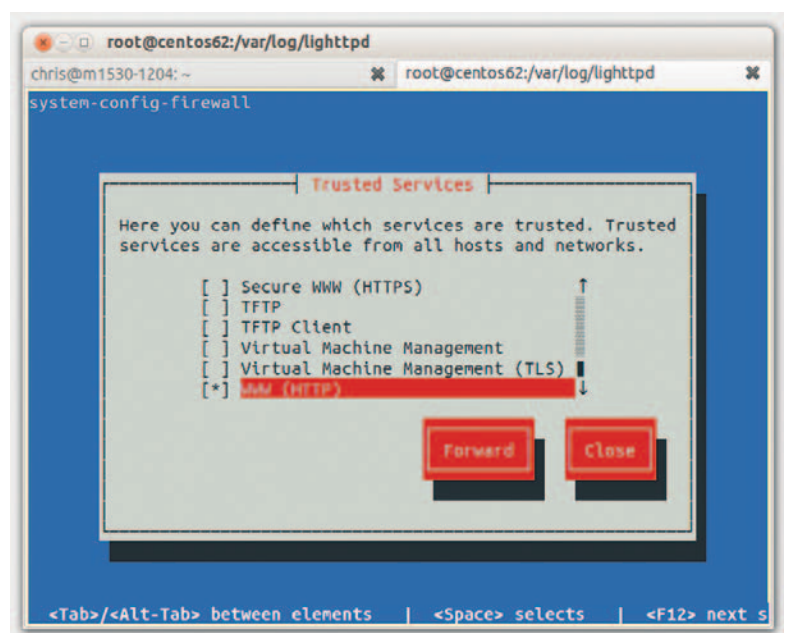
```
# lsof -c lighttpd | grep /var/log
```

Команда показывает, что это файлы `/var/log/error.log` и `/var/log/access.log`. В них нет ничего особенно интересного, но в лог-файле с ошибками часто есть сообщения, из которых можно понять, почему не работает сервис, особенно если это сервис вроде `bind` (сервер DNS), очень придирчивый к синтаксису файлов настройки и файлов зон. Вот удобный способ наблюдения за запуском сервера: откройте два терминала рядом и в первом запустите `tail -f` с именем лог-файла (так вы сможете видеть новые сообщения), а во втором – запустите демон командой `service`.

Одна из моих самых частых ошибок при решении проблем – обращение к журналам в самую последнюю очередь; на самом деле при диагностике проблемы прежде всего загляните именно в них.

В последней части небольшой серии мы наденем наши шляпы безопасности и обратимся к этой теме. Увидимся! **LXF**

» С помощью утилиты `system-config-firewall` легко открыть новые порты в брандмауэре. Эти изменения будут постоянными.



Профилирование и оптимизация

Код тормозит? Требуется оптимизация? Изучите мир статистики производительности ядра с *perf*.

Подозреваю, что в мире есть множество вопиюще недо-
загруженных серверов, время простоя которых превы-
шает 90 %. В самом деле, увеличение загрузки серверов – один из движущих причин использования виртуализации. Но на другом конце спектра есть и много систем, которые выжимают производительность из Linux до крайнего предела. Если вы пытаетесь нацедить из своих серверов еще капельку, рассмотрите утилиту мониторинга производительности под названием *perf*, способную помочь найти наиболее часто используемые участки

кода, оптимизация которых принесет плоды с наибольшей вероятностью. *Perf* – интерфейс командной строки к свежей (ну, до-
вольно свежей) возможности ядра Linux: записи параметров производительности для выполняемого кода. Некоторые из этих замеров аппаратные, они используют счетчики процессора для фиксации таких событий, как промахи кэша или для подсчета

«Не придавайте результатам много значения, если выборка мала.»

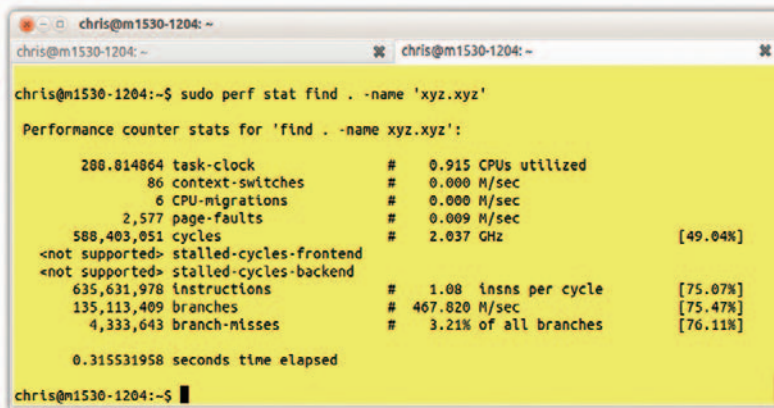
количества выполненных команд. А некоторые – программные, в них с помощью таймеров или других программных событий анализируется поведение программы.

Как узнать больше

По ссылке <http://bit.ly/LKS7rn> можно найти руководство, а по ссылке <http://bit.ly/PJ9DMY> – видео на YouTube от Доминика Дюваля [Dominic Duval] (для просмотра нужно войти в систему). Или просто загрузите *perf* и начинайте эксперименты. В Ubuntu я нашел его в пакете *linux-tools-common*; в CentOS – в пакете *perf*.

Синтаксис команд *perf* напоминает синтаксис *rpm* или *git* наличием множества субкоманд. Например, **perf list** выведет список доступных событий, **perf top** сформирует динамические данные профилирования для всей системы (нечто вроде *top*), **perf record** запишет заданные события для заданной команды (она также может подключиться к запущенному процессу), а **perf report** сформирует отчет по сгенерированному данным. С **perf annotate** можно даже спуститься до уровня машинных команд. Она запустит заданную команду, разберет каждую функцию и сформирует отчет с процентными данными для каждой выполненной команды. Но не придавайте результатам слишком много значения, если выборка мала. Помните шутку о лжи, гнусной лжи и статистике?

После запуска *perf* у вас появится множество цифр, как после *vmstat* или **cat /proc/interrupts**; и уж будьте уверены, они вам о чем-то хотят сообщить, нужно только знать – что. Думаю, следует написать руководство с примерами. Есть добровольцы?



```
chris@m1530-1204:~$ sudo perf stat find . -name 'xyz.xyz'
Performance counter stats for 'find . -name xyz.xyz':

   288,814864 task-clock           #    0.915 CPUs utilized
         86 context-switches      #    0.000 M/sec
          6 CPU-migrations        #    0.000 M/sec
    2,577 page-faults            #    0.009 M/sec
588,403,051 cycles                #    2.037 GHz                    [49.04%]
<not supported> stalled-cycles-frontend
<not supported> stalled-cycles-backend
 635,631,978 instructions        #    1.08 insns per cycle        [75.07%]
 135,113,409 branches            # 467.820 M/sec                 [75.47%]
 4,333,643 branch-misses        #    3.21% of all branches       [76.11%]

0.315531958 seconds time elapsed

chris@m1530-1204:~$
```

› Статистика деятельности ядра, собранная *perf*. Знать бы еще, к чему бы эти цифры...

Что делает программиста хорошим?

На своих курсах по программированию я вижу много одаренных, опытных разработчиков, которые ясно «понимают» программирование и хотят изучать новые языки или фреймворки. Они пишут код быстро и уверенно, и мне часто очень нравится стоять у них за спиной и смотреть, как появляется код. Есть и новички, которые испытывают сложности в конкретных вещах, но у которых все получится, когда они поймут разницу между **argv++* и *(*argv)++*. Но я вижу и грустные случаи: неудачников, у которых никогда не получится зарабатывать на жизнь написанием программ.

Поэтому я задал себе вопрос: «Что делает хорошего программиста хорошим?» Дональд Кнут [Donald Knuth] – ушедший на покой, но еще очень активный профессор Стэнфордского университета. Его самая известная работа – многотомник «Искусство программирования» (если вы не слышали о нем, наберите это название в Google). Кнут писал: «Психологическое свойство программистов – главным образом, возможность перемещаться по уровням абстракции с низкого на высокий. Видеть что-то в мельчайших деталях и в целом». Я бы согласился с его высказыванием.

Наука или искусство?

Еще одна черта хороших программистов, которую заметил лично я – постоянная и ненасытная потребность в новых знаниях, будь то новый язык или мелкая особенность системного вызова *fcntl()*. И третье – они считают свое занятие искусством. Снова цитирую Кнута: «Наука – это то, что мы понимаем достаточно хорошо, чтобы объяснить компьютеру. Искусство – все остальное, что мы делаем». Интересно, что думаете вы? Что делает хорошего программиста хорошим? Программирование – это искусство или наука? Пишите мне на chris.linuxformat@gmail.com.

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

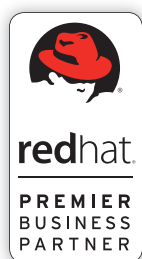
Сертифицирована ведущими производителями оборудования и разработчиками ПО



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

Итак, вы хотите стать сисадмином?

Шестая часть серии, которая превратит вас из новичка в звезду системного администрирования. Поговорим о безопасности.

В этой, последней, части серии я хочу обратиться к теме безопасности серверов. Наверное, оставить ее напоследок было ошибкой, поскольку можно подумать, что безопасность – это то, чем можно заняться в последнюю очередь, когда все остальное уже сделано. Так поступать не стоит. Безопасность и разумный уровень паранойи должны быть заложены во всех ваших действиях. Кстати, если вам покажется, что большая часть того, о чем мы поговорим сегодня – нечто само собой разумеющееся, для этого есть причина. Для большинства из этого.

Для начала попытаемся дать определение безопасности. Формальная модель безопасности описывается моделью CIA – конфиденциальность (Confidentiality), целостность (Integrity) и доступность (Availability).

Конфиденциальность – это гарантия того, что информация на ваших серверах доступна только тем, у кого есть право ей пользоваться. Целостность – это гарантия того, что информация сохраняется в первоначальном виде – это относится не только к данным, но и к программам. А доступность – это гарантия того, что данные и сервисы, предоставляемые системой, доступны именно должным образом (например, можно удовлетворить два первых требования, отключив компьютер от сети и заперев его в сейф, но это не слишком хорошо повлияет на доступность данных; помните, что атаки, нацеленные на отказ от обслуживания и призванные неблагоприятно повлиять на доступность вашей системы – одна из самых популярных разновидностей вредоносных атак).

Безопасность – это не программа, которую можно установить. Это постоянный процесс. Поэтому вот план поддержки безопасности системы из десяти пунктов.

1: Вовремя загружайте обновления безопасности своего дистрибутива

Время от времени в приложениях выявляются потенциальные уязвимости. У крупных дистрибутивов Linux есть сервисы обновления, с которых можно загрузить обновления безопасности для исправления этих ошибок в течение периода поддержки дистрибутива. Для RedHat 6 это 10 лет (но нужно приобрести подписку). В предыдущих версиях RedHat для применения заплат использовалась утилита *up2date*, но от нее отказались в пользу *yum*; благодаря модулю расширения *yum-plugin-security*, *yum* сможет вывести список обновлений безопасности и установить их.

Другой пример – релизы Ubuntu Server с долгосрочной поддержкой [Long Term Support], например, 12.04 LTS, бесплатны и поддерживаются в течение пяти лет. В Ubuntu для автоматической установки обновленных пакетов можно воспользоваться пакетом *unattended-upgrades* – в нем можно настроить обновление всех пакетов или только обновлений безопасности.

2: Выбирайте надежные пароли

Любая система, у которой есть внешнее интернет-подключение и сервис аутентификации (например, ssh), ежедневно подвергается тысячам попыток несанкционированного входа в систему от автоматизированных скриптов. Такова жизнь.

Выбор надежных паролей для учетных записей пользователей жизненно необходим. Я писал об этом недавно (см. **LXF160**) и не буду повторяться. Но не забудьте о паролях на маршрутизаторах и других сетевых устройствах – никогда не оставляйте их в значениях по умолчанию.

3: Регулярно делайте резервные копии

Возможно, вы думаете, что между резервным копированием и безопасностью нет прямой связи. Но помните, что одна из составляющих CIA – доступность. Поэтому если плохие парни уничтожат ваши данные (или если ваш незапаянный диск испустит последний вздох и скончается), резервные копии имеют решающее значение для возобновления доступности данных.

Существует множество утилит резервного копирования на разные случаи жизни. В простейшем случае воспользуйтесь *rsync* для эффективного переноса своей файловой системы на удаленный

Это не наша политика

Если вы системный администратор Linux, то ваша жизнь будет гораздо проще, если у компании есть политика безопасности. Это обширная тема, и мне не хочется начинать составлять политику за вас.

Политика может определять, у кого должен быть физический доступ к серверам, каковы минимальные требования к паролям («все пароли должны состоять из букв, цифр, каракулей, жестов и пороссячьего визга»), при каких обстоятельствах можно отключать сервер от сети, как очищать диски перед их утилизацией и т. д.

Если вам нужны примеры, зайдите на *infotech.com* или на сайт SANS (<http://bit.ly/SvgZrk>).



➤ Вам помогут оставаться в безопасности следующие программы (по часовой стрелке сверху): *selinux*, *ssh*, *clamav*, *shorewall*, *sans* и *arparmor*.

Вы слышали это и раньше...

Если у вас есть подписка на **LXF** и доступ к электронной версии или старые номера журнала, можете вернуться к нескольким ранним статьям, посвященным безопасности.

LXF108: AppArmor

LXF115: Утилиты брандмауэра

LXF139: `ufw` («простой брандмауэр» Ubuntu)

LXF148: Повышение привилегий (`su`, `sudo` и т.д.)

LXF158: SELinux

LXF160: Пароли и политики

компьютер, или создайте архив с файлами на съемном носителе с помощью `tar`. Если вы это делаете, помните о том, от чего защищаетесь. Я регулярно делаю архивы своей личной файловой системы на внешний жесткий диск с помощью `tar`, но, к своему стыду, ношу этот диск в сумке для ноутбука вместе со всем остальным. Так я защищаюсь от неисправности жесткого диска, но не от кражи сумки для ноутбука.

Если у вас несколько серверов, вас могут заинтересовать средства централизованного резервного копирования, такие как *BackupPC* или *Vacula*, с помощью которых можно делать резервные копии нескольких систем на один компьютер. Для настройки обеих потребуются некоторые усилия. Подумайте о том, где будете хранить резервные копии. Лучше всего хранить их в другом месте (это поможет, если ваш дом сгорит дотла), но это неудобно, если нужно быстро восстановить данные. Также нужно доверять тем, кто будет хранить или шифровать их. Скажем, многие хранят их на облаке (например, в Amazon S3), это позволяет быстро восстановить данные и стоит порядка 10 центов за 1 ГБ в месяц.

Недавно Amazon запустила новый сервис долговременного архивирования Glacier, время получения данных в котором может составлять до нескольких часов. Расценки начинаются от 1 цента за ГБ в месяц. Разумеется, для резервного копирования в облако нужен хороший исходящий канал.

4: Не запускайте сервисов, которые вам не нужны

Это очевидно. Если вам не нужно предоставлять клиентам, скажем, FTP, не запускайте сервер FTP. Лучше не устанавливайте его вообще. Не то что он небезопасен, но в серверах FTP определенно были обнаружены уязвимости, позволяющие получить доступ `root`. Зачем рисковать? Это было проблемой ранних версий Unix, в которых все сервисы по умолчанию были запущены и приходилось отключать большинство из них (конечно, и времена тогда были проще). В современных дистрибутивах Linux по умолчанию (обычно) включен только `ssh`.

5: Обучайте своих пользователей

Понимаю, что если мы говорим о серверах, трудно понять, что имеется в виду под «пользователем». В данном контексте это некто, обладающий информацией, которая может помочь злоумышленникам атаковать вашу систему, или тот, у кого есть учетная запись на настольном компьютере, подключенном к сети. Получение доступа к такому компьютеру – прекрасное начало для злоумышленника, пытающегося подобраться к корпоративной базе данных. Конечно, было бы здорово избавиться от ненадежных компьютеров с Windows и заменить их на компьютеры с Linux, но это потребует времени. Поэтому как минимум убедитесь, что пользователи знакомы с основами: не открывать вложения в письмах от незнакомых людей; блокировать компьютер, уходя на обед; не сообщать никому свой пароль; и не посещать сайты, не имеющие отношения к работе.

У SANS есть несколько отличных ресурсов, которые помогут в обучении пользователей: www.securingthehuman.org и www.sans.org. И если у вашей компании нет программы обеспечения безопасности, подумайте, как убедить своего босса начать ее.

6: Читайте журналы (или хотя бы отчеты)

Ну да, лог-файлы неподдельно скучны, и вряд ли кто-то будет их читать. Но есть утилиты, например, `logwatch`, которые прочитают их за вас и сгенерируют отчеты. Например, команда

```
# logwatch --detail med --range Today --print
```

сформирует отчет о сегодняшних логах. В дистрибутивах *logwatch* часто включена и настроена по умолчанию. Обычно она запускается как ежедневное задание *cron* и отправляет отчеты на почту `root`. Но пользы от нее никакой, если не заглядывать в отчеты. Что в них следует искать? Сообщения о добавленных учетных записях, которые вам незнакомы, о новых (неизвестных) установленных программах и о повторяющихся ошибках входа в систему от таких программ, как `sshd` и `su`.

7: Не отключайте средства мандатного контроля доступа

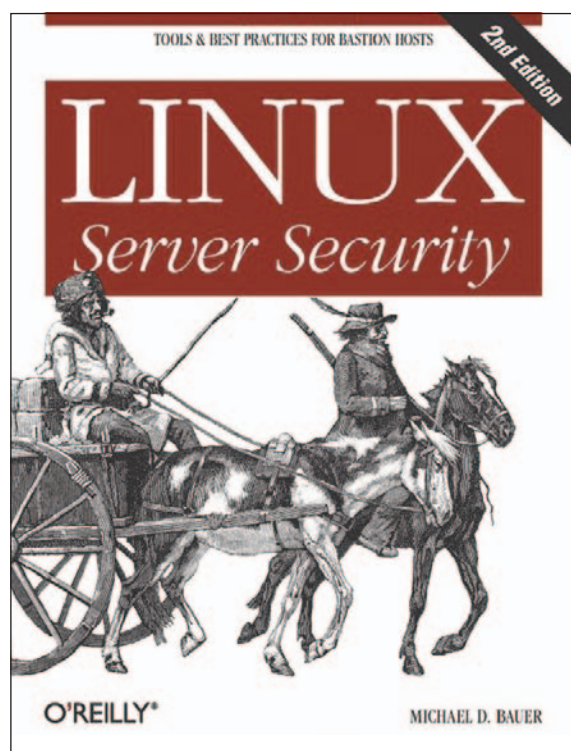
Если вы не поняли, о чем я, то под ними я имею в виду такие вещи, как SELinux и AppArmor. Несколько раз пострадав от SELinux (как я на моих учебных курсах), есть соблазн его отключить. Если что-то не работает, можно временно отключить SELinux, чтобы узнать, в нем ли дело, хотя если ваш уровень паранойи достаточно высок, вы не захотите этого делать. Но отключать его совсем – явно не лучшая идея. У него есть свои задачи (если у вас есть старые номера, загляните в **LXF158**, там я немного писал об этом).

С помощью утилиты *audit2allow* можно создать правила SELinux для приложений, у которых их нет. Месяц назад я рассказывал об этом. Для систем с AppArmor есть утилита *aa-genprof*.

8: Изучите и безопасно настройте запускаемые приложения

Здесь трудно сказать что-то конкретное: все зависит от приложений. Вот несколько примеров. Во-первых, для демона *ssh*

»



» Замечательная книга Михаэля Бауэра [Michael Bauer] продолжит наш сегодняшний рассказ. Она немного старовата (2005), но я не знаю ничего новее.

можно отключить вход в систему от имени суперпользователя root, что заставит пользователей использовать обычные учетные записи и при необходимости переключаться на root с помощью su. Убедитесь, что он настроен именно так (посмотрите значение параметра PermitRootLogin в `/etc/ssh/sshd_config`). Другой пример – в *Samba* можно ограничивать диапазоны IP-адресов, в которые она экспортирует разделяемые каталоги. Убедитесь, что эти диапазоны покрывают вашу сеть и ничего кроме.

9: Настройте брандмауэр

Запустить брандмауэр с фильтрацией пакетов в Linux просто, благодаря коду сетевого фильтра, который встроен в ядро, и команде `iptables`, которая управляет правилами. Однако если вы новичок в Linux, я не советовал бы запускать `iptables` напрямую. Наборы правил для настоящего брандмауэра могут быть очень сложными и требуют глубоких знаний TCP/IP.

В RedHat пользуйтесь утилитой `system-config-firewall`. С ней легко открыть нужные порты; также можно задать «доверенные» интерфейсы, которые не будут фильтроваться (например, для внутренних сетей).

В Ubuntu есть небольшая утилита `ufw`, которая с помощью довольно простого (по сравнению с `iptables`) синтаксиса позволяет управлять правилами. Например, для открытия порта достаточно скопировать

```
$ sudo ufw allow ssh
```

А если хотите сделать кое-что похитрее, можно сделать следующее:

```
$ sudo ufw allow proto tcp from 192.168.0.0/24 to any port 22
```

Так вы получаете больше возможностей по сравнению с утилитами вроде `system-config-firewall`.

С помощью довольно удачно названной графической программы `firestarter` (www.fs-security.com) можно легко задавать правила и просматривать трафик, который блокирует брандмауэр. Стоит взглянуть и на `shorewall` (www.shorewall.net). Это еще одна утилита, которая генерирует наборы правил `iptables` из описания поведения брандмауэра на языке высокого уровня.

Мир *Shorewall* – это не щелчки мышью, а текстовые конфигурационные файлы. В них можно задать определенные «зоны» и оговорить движение трафика между ними. Это именно то, что требуется для настройки шлюза, у которого есть сетевые соединения с внешним миром, с внутренней частной сетью и с «демилитаризованной зоной» публично доступных серверов.

10: Регулярно запускайте средство обнаружения вторжений

Принцип работы средств обнаружения вторжений обычно состоит в том, что они делают «снимок» файловой системы (или указанных вами частей) в исходном, чистом состоянии компьютера, а затем периодически делают новые снимки и сравнивают их с оригиналом. Снимок – не полная копия файлов, он содержит лишь такие параметры, как размер, владельца, время доступа и контрольную сумму MD5 каждого файла. Этого обычно достаточно для распознавания, что что-то было изменено.

Обычно подобным образом нужно отслеживать только системные каталоги, например, `/etc`, `/bin` и `/usr/bin`, все изменения в которых должны быть вам известны. Конечно, такие утилиты скорее позволяют среагировать на проблему, чем предотвратить ее. Они не мешают хакеру взломать вашу систему, но вы по крайней мере об этом узнаете.

Классическая утилита для этих целей – `tripwire`, и ее открытая версия все еще доступна (<http://bit.ly/h2pr9>). Однако полноценная версия уровня предприятия стала коммерческим продуктом от Tripwire, Inc. Еще одна коммерческая утилита – `Cimtrak`. Также можете взглянуть на AIDE (Automatic Intrusion Detection Environment – среда автоматического определения вторжений),

Под замком

Если вы не задали пароль для BIOS, помните, что каждый, у кого есть физический доступ к серверу, сможет загрузить его с собственного носителя и получить полный доступ. При задании пароля BIOS задача усложняется лишь немного, разве что теперь злоумышленнику понадобится отвертка, чтобы снять жесткий

диск. Нужно ограничить физический доступ к серверу и повесить физический замок или установить какую-то систему с электронными картами. Каждый раз, когда я прихожу читать курсы в компанию с большим дата-центром, я всегда прошу показать мне серверную и почти всегда слышу: «Извините, нет».

которая скорее всего есть в репозиториях вашего дистрибутива. Учтите, что эти утилиты нужно настраивать, и отчеты, как и лог-файлы, бесполезны, если их не читать.

Если вы пришли из мира Windows, то, наверное, интересуетесь, почему я не предложил вам установить антивирус (один из моих сыновей на своей последней работе управлял инфраструктурой крупного программного проекта и проводил, кажется, половину своего времени за установкой, обновлением и запуском антивирусных программ). Вредоносное ПО в Linux представляет собой гораздо меньшую проблему – частично, подозреваю, потому, что Linux гораздо реже становится целью атак плохих парней, а также из-за более высокого уровня безопасности.

Но антивирусные продукты для Linux, разумеется, существуют. Популярное открытое решение – программа `ClamAV`. Есть и коммерческие продукты, например, `Avast`. Большинство вирусных сигнатур, которые они ищут – это вирусы для Windows, поэтому запуск таких утилит в Linux защищает компьютеры в основном от неблагоприятной роли «переносчиков», т.е. от наличия в них инфицированных файлов, которые могут быть переданы на компьютеры с Windows.

Надеюсь, что эта статья не превратила вас в маниакальных параноиков. Безопасность – это во многом просто здравый смысл. Не забывайте, что однажды сказал президент Эйзенхауэр: «В напрасных поисках абсолютной безопасности мы станем банкротами». LXF



➤ Безопасность повысит шифрование сетевых подключений. Хорошо, что это больше не надо делать вручную.

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Упомянул PostgreSQL на защите кандидатской диссертации по физике элементарных частиц.

Ричард Столлман нагнетает

Произносить слова мы научились. Теперь бы научиться считать деньги.
Виктор Степанович Черномырдин

Жил-был простой и незатейливый, в чем-то добрый, в чем-то хитрый парень Марк. Слетал Марк в Космос и решил подарить человечеству кусочек общего достояния, подкрепленного южноафриканским колоритом. Рассылал Марк диски счастья по людям, а люди их ставили, причем некоторые радовались. В общем и целом, польза человечеству выходила.

К сожалению, с дарением счастья всегда возникает коварный вопрос: а кто оплатит этот банкет щедрости? Люди не любят делиться ресурсами – это нормально; и поэтому часто в качестве решения приходит принуждение к оплате в той или иной форме. Пока в данном случае от этой чести можно отказаться, но искушение будет только нарастать.

О чем это я? Да о набившем уже оскомину поиске по товарам в Amazon в грядущей версии Ubuntu. Ричард Столлмен, как и всегда, взял на себя роль Капитана Очевидность и назвал шпионское программное обеспечение шпионским же программным обеспечением. То, что он не любит подобного рода сюрпризы, всем давно известно, и неудивительно, что Ubuntu теперь угодила у него в опалу.

Безусловно, любой человек имеет неотъемлемое право поставить себе чужое шпионское программное обеспечение, ведь это обеспечение бывает таким удобным. Просто печально, что значимая часть движения СПО этому содействует.

e.m.baldin@inp.nsk.ru

В этом месяце вы научитесь...



Пересылать деньги 68

Козн Вервлоесем по горло сыт центробанками, так что решил вместо них пользоваться цифровой валютой. Узнайте, как это делается.



Создавать web-приложения 72

Джоно Бэкон познакомит вас с фреймворком, который поможет ускорить разработку web-приложения вдвое.



Запускать Geany 76

Любой программист настолько хорош, насколько хороша его IDE, и **Джоно Бэкон** решил, что он откопал самую гениальную.



Осваивать udev 80

Мантра UNIX – «Все на свете является файлом»; ну, а как приложить ее к аппаратуре? **Нейл Ботвик** разбирается.



Отгонять шпионов 82

Нейл Ботвик терпеть не может, когда подслушивают, и применяет SSH, чтобы подслушивать было нелегко.



Автоматизировать дом 84

Козн Вервлоесем позволил своему компьютеру царить в доме, и обнаружил, что это очень даже неплохое решение.



Строить Pi (дистрибутив) 88

Джон Лэйн решительно намерен владеть своим Raspberry Pi безраздельно, и построил специальный дистрибутив лично для себя.



Вникать в Erlang 92

Андрей Ушаков построил многозадачные функции, которые обрабатывают порции исходных данных параллельно.

Пропустили номер?

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



БИТКОЙН: Валюта

Козн Вервлоесем пояснит, как управлять своими финансами через одноранговую валюту, обходящуюся без централизованного контроля.



Наш эксперт

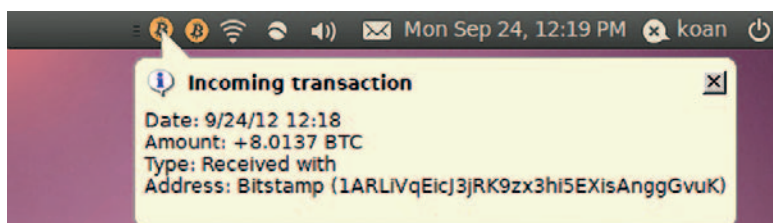
Козн Вервлоесем пишет об открытии в 2000 года, и испытывает симпатию к технологиям, дающим людям силу и власть.



Осуществляя платежи онлайн через PayPal или онлайн-сервис своего банка, вы полагаетесь на милость провайдера платежа или банка. Если они сочтут вашу транзакцию подозрительной, то могут заблокировать ваш счет целиком. И что, вы продолжаете считать себя полным хозяином своих денег? Подумайте еще раз. Например, в конце 2010 года PayPal заморозила счет Wikileaks, и есть и другие многочисленные примеры малых компаний, чьи счета на PayPal с многомесячным доходом были заморожены на неопределенный срок.

Более того, если взглянуть на деньги с более общей точки зрения, вы даже не в силах управлять ценностью своих денежных средств. В ответ на финансовый кризис, Британский, Американский и Европейский центробанки «печатают» все больше и больше денег. В соответствии с законом спроса и потребления,

» Транзакции Биткойн совершаются мгновенно, даже в выходные и в нерабочее время!



возросшая денежная масса означает, что средства, которые вы копили годами, теряют стоимость...

Однако альтернатива есть: это Биткойн. Она разработана загадочным человеком по имени Сатоши Накамото [Satoshi Nakamoto], который в 2009 году опубликовал документ под названием "Bitcoin: A Peer-to-Peer Electronic Cash System [Битовая монета: одноранговая система электронной валюты]". Согласно аннотации к его статье, его намерение заключалось в создании «версии чисто одноранговой системы электронной валюты, которая позволила бы перечислять платежи напрямую от одной стороны к другой без привлечения каких-либо финансовых учреждений». Поскольку вам не нужны финансовые учреждения для хранения или передачи вашей электронной валюты биткойн, никто не сможет заморозить ваш «счет». Зато вы сможете переводить биткойны в любое место в мире за считанные минуты и незначительные отчисления. С традиционными деньгами такое невозможно.

Нарыть себе денежек

Главнейшее различие между Биткойн и почти любой другой валютой – отсутствие централизованного управления. Здесь нет центрального агентства или казначейства, которое пыталось бы сбалансировать инфляцию, зато и общее количество биткойнов, которое может быть создано (21 миллион), и скорость создания биткойнов известны заранее, и ими невозможно манипулировать. Биткойны генерируются компьютерами, являющимися частью сети, на которых работает «добывающее ПО [mining software]», помогающее провести и утвердить транзакции биткойн. Проведение этих транзакций задействует некоторые ресурсы CPU, и в качестве вознаграждения за эту работу и, следовательно, за обеспечение работы всей сети Биткойн, каждый десять минут создаются новые биткойны и выдаются одному из таких «добытчиков [miner]».

«К счастью, биткойны, как и любую валюту, можно купить.»

И все это основано на хитроумном шифровании, выполненном одноранговым способом. Например, каждый пользователь, у кого работает клиент Bitcoin, имеет копию всех когда-либо выполненных транзакций Биткойн, поэтому клиент может проверить, является ли транзакция подлинной или поддельной. Будучи владельцем «монеты [coin]», вы передаете ее кому-то другому, подписывая своим приватным ключом хэш предыдущих транзакций и публичный ключ следующего владельца, и добавляете все это в конец цепочки подписей. Тот, кому вы заплатили, может проверить предыдущих владельцев, удостоверяя все подписи вплоть до первого владельца монеты, добытчика. Дважды потратить один и те же биткойны также невозможно из-за все тех же проверок. Довольно теории; потранжируем-ка цифровые деньги. И вот первая проблема: как же нам, собственно, биткойны получить?

цифровая

Их добыча не представляется делом выгодным, если только под вашим письменным столом не скрываются лошадиные силы для перемалывания чисел; но, к счастью, биткойны, как и любую валюту, можно купить. Для этого и существуют биржи: вы создаете учетную запись на бирже Биткойн, вносите туда определенное количество денег в любой валюте и конвертируете их в биткойны.

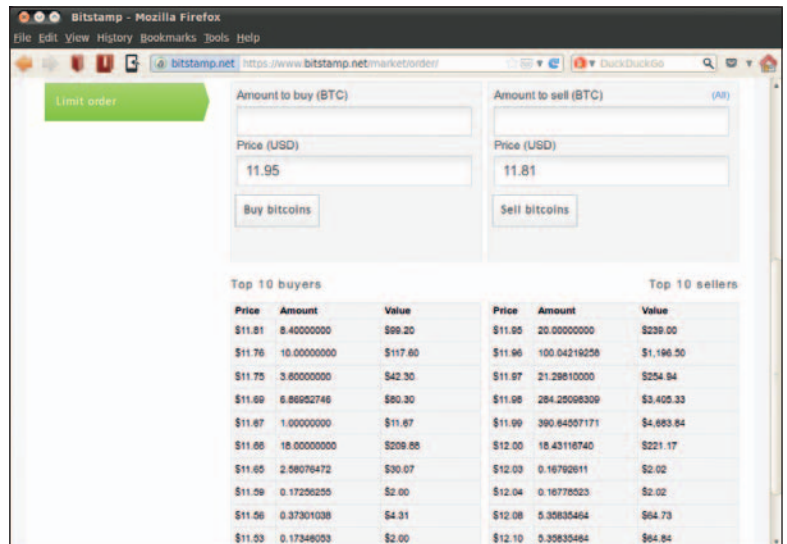
Самая популярная биржа Биткойн на данный момент – Mt.Gox, базирующаяся в Японии, затем идут Intersango (Великобритания) и Bitstamp (ЕС). На самом деле совершенно не важно, где расположена биржа Биткойн, просто у ее расположения есть вполне практические последствия – например, сколько дней уйдет на пополнение вашего счета. Биржа показывает вам книгу заказов с предложениями (продавцов биткойнов) и запросами (покупателей биткойнов), и можно увидеть, что вы получите за свои деньги. Обменный курс может значительно варьироваться в зависимости от степени доверия к стабильности биткойна, так что до приобретения биткойнов стоит ознакомиться с текущими тенденциями. Вы можете немедленно купить биткойны по текущей цене предложения, либо установить лимитную заявку: это заявка на приобретение биткойнов, когда цена опустится ниже установленной величины. Когда вы, наконец, обзаведетесь биткойнами, вы сможете перевести их со своего счета на бирже на собственный адрес Биткойн, о чем мы расскажем вам ниже.

Клиент Satoshi

Официальный клиент Биткойн, который можно загрузить с www.bitcoin.org, называется *Bitcoin-Qt*, но пользователи Биткойн именуют его клиентом *Satoshi*, потому что именно его изначально разработал Сатоши Накамото перед своим таинственным исчезновением. Текущий релиз – 0.7.0. Программа показывает вам ваш биткойновый кошелек, на котором изначально находится 0 BTC (BTC – символ валюты Биткойн). После запуска клиент *Satoshi* начинает синхронизироваться с сетью. Для самого первого раза это означает, что он скачивает все существующие блоки транзакций, что может занять пару часов. Теперь щелкните по Receive Coins [Получение денег]. Это окно покажет вам ваш адрес Биткойн



» В прошлом цена биткойна очень резко менялась, так что не конвертируйте все свои сбережения в BTC!



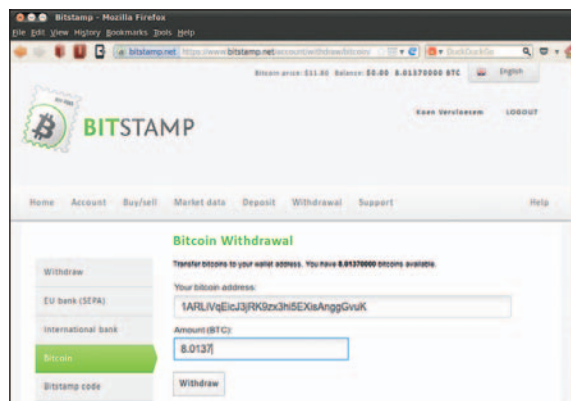
по умолчанию для получения платежей. Адрес Биткойн (например, такой: 1koevXPspX9aYwsT1MocYQNB9EXoQwLi) аналогичен номеру банковского счета: его можно использовать для перевода на него денег. Но разница в том, что в Биткойн вы сами можете создавать столько адресов, сколько захотите. Обычно новый адрес создается (нажатием на кнопку New Address) для каждого, от кого вы ожидаете получить платеж, чтобы легко отслеживать, кто вам заплатил. Еще одно преимущество – ваша анонимность, поскольку одноранговая природа Биткойн означает, что баланс каждого адреса Биткойн в полной мере публичен. Так что если вы передаете тем, кто вам должен заплатить, один и тот же адрес, они смогут увидеть, сколько вам уже платили другие.

Итак, создайте новый кошелек и пометьте его названием биржи, где хранятся ваши биткойны. Откройте свою учетную запись на сайте биржи, и откройте страницу, чтобы перевести свой баланс Биткойн на адрес своего кошелька. Перейдите в клиент

» Покупая биткойны, используйте лимитную заявку, чтобы попытаться сделать это по цене ниже текущей цены покупки.

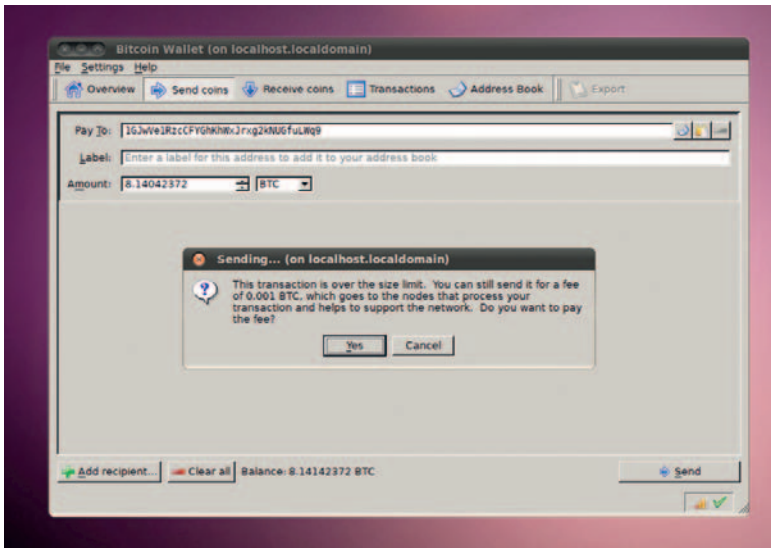
Скорая помощь

Можно найти информацию о любом адресе или транзакции Биткойн, введя адрес или ID транзакции в строку поиска сайта <http://blockchain.info>.



» Переведите приобретенные вами биткойны на свой собственный кошелек с помощью Bitstamp.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



➤ Ни один банк не переберет взноса размером 0,001 BTC, или менее 1 пени, за моментальный международный перевод ваших денежных средств.

Satoshi, щелкните правой кнопкой по адресу и выберите Copy Address [Копировать адрес], вставьте его в поле адреса Биткойн на сайте вашей биржи и укажите количество биткойнов, которое вы хотите перевести. После нажатия на Withdraw [Забрать] пройдет меньше минуты, и вы увидите, как перевод поступает в ваш клиент *Satoshi*. Только крупные переводы обрабатываются вручную на бирже, что занимает больше времени.

Ну вот, в вашем кошельке виртуально забренчали биткойны; как же их тратить? Первый вопрос – а что продается за биткойны? На сайте <https://en.bitcoin.it/wiki/Trade> имеется весьма полный список, в котором найдется все – от торговцев серебром и золотом до сайтов аукционов и азартных игр, интернет-сервисов и даже ресторанов и отелей. Если вы решите купить что-то на одном из этих сайтов, или если вы решили заплатить кому-то в биткойнах, получатели вашего платежа дадут вам свой адрес Биткойн (или, более точно, один из их адресов Биткойн).

Затем откройте клиент *Satoshi*, щелкните по Send Coins [Отправить деньги], вставьте адрес, куда надо перевести ваш платеж, в поле Pay To [Кому заплатить], добавьте ярлык для ссылки на будущее, укажите количество биткойнов и нажмите на Send. Иногда вам придется оплатить скромный взнос, если размер платежа слишком велик. Если все прошло хорошо (и ваш баланс Биткойн достаточно велик!), другая сторона практически немедленно увидит, как проходит транзакция. Но учтите: все транзакции Биткойн по своей природе необратимы!

«Вы сами отвечаете за безопасность ваших денег: тут вам не банк.»

Вкладка Overview [Просмотр] показывает ваши недавние транзакции Биткойн, как в качестве отправителя, так и в качестве получателя, и вы можете увидеть все свои транзакции во вкладке Transactions. По двойному щелчку на транзакции в последнем окне клиент *Satoshi* покажет вам подробности. Интересная деталь – поле Status, указывающее количество «подтверждений», то есть узлов Биткойн, которые подтвердили действительность транзакции. Чем выше это число, тем меньше шансов, что транзакция была подделана. Общее правило гласит, что вам нужно дождаться шести подтверждений, чтобы удостовериться в действительности транзакции. По этой же причине клиент *Satoshi* показывает вам неподтвержденную часть вашего баланса отдельно, во вкладке Overview.

Подпись и верификация сообщений

Клиент *Satoshi* имеет еще одну интересную функцию: подпись и удостоверение сообщений. Подпись сообщения с помощью адреса Биткойн в качестве ключа – это способ доказать, что сообщение пришло от владельца данного адреса Биткойн, поскольку это делается в клиенте *Satoshi* с помощью приватного ключа, принадлежащего данному адресу. Допустим, вы заказали на сайте продавца несколько золотых монет. Вы платите продавцу с помощью своего адреса Биткойн, и он просит вас подписать сообщение этим адресом, указав, куда вы хотите доставить свой заказ.

Итак, щелкните в меню File [Файл] по Sign Message... [Подписать сообщение], вставьте или выберите нужный адрес и введите свое сообщение. Будьте как можно более точны, указывая, кто вы, в том числе свой ID покупателя или зарегистрированный у продавца адрес электронной почты, адрес для доставки, имя продавца и ID вашей транзакции. Нажмите на кнопку Sign Message,

скопируйте полученную в результате подпись и отправьте и сообщение и подпись (например, электронной почтой) продавцу. Он откроет свой клиент *Satoshi*, откроет меню File > Verify Message... [Файл > Проверить подлинность сообщения], вставит ваш адрес Биткойн, сообщение и подпись в соответствующее поле и нажмет на кнопку Verify Message, чтобы проверить, действительно ли это вы просите его доставить недешевые золотые монеты по указанному адресу.

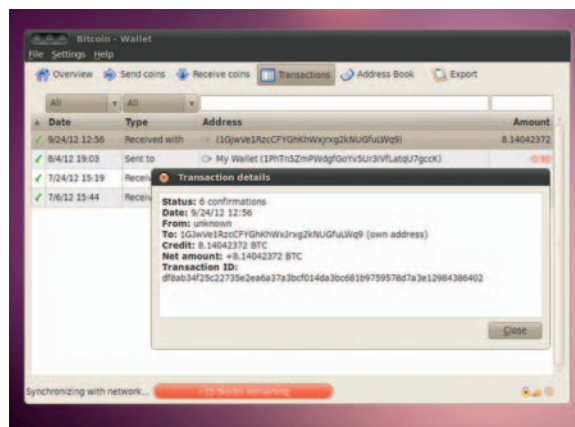
В прошлом году было немало плохих новостей о Биткойн. Их причина не в том, что безопасность Биткойн недостаточна, а в том, что большинство людей, и даже многие сервисы Биткойн, не слишком серьезно относятся к безопасности. Основное отличие Биткойн от традиционной валюты в том, что за безопасность ваших денег отвечаете вы сами: тут вам не банк, который сделает это за вас.

В прошлом году было немало плохих новостей о Биткойн. Их причина не в том, что безопасность Биткойн недостаточна, а в том, что большинство людей, и даже многие сервисы Биткойн, не слишком серьезно относятся к безопасности. Основное отличие Биткойн от традиционной валюты в том, что за безопасность ваших денег отвечаете вы сами: тут вам не банк, который сделает это за вас.

Шифрование

Во-первых и в-главных, свой кошелек Биткойн вы должны зашифровать. Если вы используете клиент *Satoshi*, то ваш кошелек хранится в `~/bitcoin/wallet.dat`. Всякий, кто получит доступ к этому файлу, сможет потратить ваши денежки. К счастью, клиент *Satoshi* имеет простую в использовании опцию шифрования кошелька. Просто щелкните по Settings > Encrypt Wallet... [Параметры > Зашифровать кошелек], затем вам нужно будет выбрать парольную фразу, с помощью которой вы зашифруете свой кошелек. Вам придется вводить эту фразу каждый раз, когда вы решите потратить свои биткойны.

Если кто-то украдет ваш незашифрованный кошелек Биткойн, он сможет преспокойно забрать ваши биткойны; но при краже зашифрованного кошелька Биткойн воры увидят баланс вашего кошелька и заглянут в транзакции, но не смогут потратить



Скорая помощь

Вы можете приобрести физические биткойны на www.casascius.com. Это настоящие монеты, которые могут храниться у вас, со встроенным приватным ключом и защитной голограммой. Вы можете потратить средства Биткойн, импортируя ключ в клиент Биткойн или отдавая физические монеты.

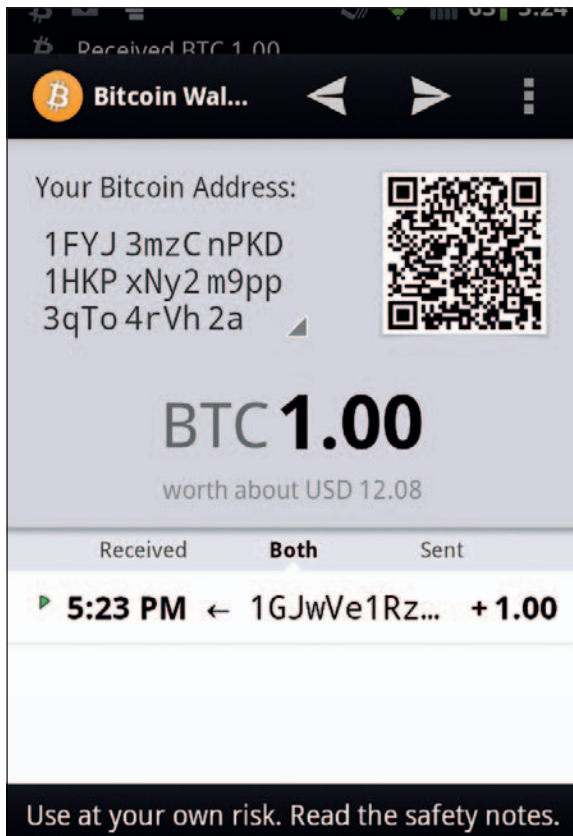
➤ По каждой транзакции вы можете проверить, сколько узлов ее подтвердили.

ни одного биткойна, не зная фразы-пароля. Причина в том, что кодовая фраза шифрует приватный ключ, принадлежащий вашему адресу и необходимый для подписи новой транзакции. Естественно, безопасность зашифрованного кошелька зависит от используемой фразы, так что вводите длинную фразу, которая не является общеизвестным предложением. Все новости о том, что биткойны у пользователей воруют с помощью вредоносного ПО, объясняются исключительно тем, что пользователи не шифруют своих кошельков или используют плохой пароль. Еще одна важная мера безопасности в Биткойн – это резервные копии. Потеряв свой файл **wallet.dat**, вы теряете весь свой баланс Биткойн, так что постоянно делайте резервные копии своего (зашифрованного!) файла **wallet.dat**.

Биржи

Еще одна проблема безопасности заключается в том, что не все биржи Биткойн подходят к безопасности со всей необходимой серьезностью. Например, в 2011 году некто взломал базу данных счетов биржи Mt.Gox и перевел 25000 биткойн с разных счетов со слабыми паролями. Конечно, нельзя быть полностью уверенным в безопасности той биржи, которую вы используете, но зато вы можете задать сильный пароль для своего счета. Более того, не стоит использовать один и тот же пароль на нескольких биржах Биткойн, чтобы исключить такую возможность, когда при взломе одного из ваших счетов все остальные тоже оказываются под угрозой.

Чтобы еще сильнее усложнить возможность взлома, несколько бирж Биткойн поддерживают двухфакторную аутентификацию. Например, Mt.Gox поддерживает в качестве второй меры аутентификации YubiKey. Если вы ее включаете, хакеру нужно не только узнать ваш пароль, но и получить физический доступ к вашему устройству YubiKey, чтобы получить шанс вломиться на вашу учетную запись Mt.Gox. Bitstamp также поддерживает двухфак-



▶ Приложение *Bitcoin Wallet* предлагает простой способ перевести другу платеж в биткойнах.



▶ Подпись сообщения своим адресом Биткойн – отличный способ подтвердить, что вы действительно автор этого сообщения.

торную аутентификацию с помощью Google Authenticator, который использует мобильное приложение на вашем смартфоне, отображающее код, вводимый при аутентификации. Но даже если вы используете сильный пароль для своей учетной записи на бирже Биткойн, и даже если у вас настроена двухфакторная аутентификация, все равно не стоит оставлять слишком много денег (будь то биткойны или традиционная валюта) на своем счете. Отправляйте деньги на свой биржевой счет, только если вам надо поменять деньги на Биткойн, и немедленно снимайте их, совершив обмен.

Другие клиенты

Клиент *Satoshi* – не единственный клиент Биткойн, но он является официальным. Поднаторев в работе с Биткойн, можете познакомиться с другими клиентами. Например, если вам не нравится медленный запуск *Satoshi*, обратите внимание на *Electrum* или *Multibit*. Оба этих клиента намного быстрее синхронизируются с сетью Биткойн, и акцентируют внимание на скорости и простоте в использовании.

Если вы считаете себя продвинутым пользователем, стоит взглянуть на *Armory*. Этот клиент зависит от *Satoshi*, который должен работать в фоновом режиме, и расширяет его множеством дополнительных функций: например, поддерживает шифрование нескольких кошельков и умеет делать резервные копии кошельков Биткойн на бумаге. Также он поддерживает транзакции оффлайн: у вас есть кошелек со на компьютере, работающем оффлайн, полностью неуязвимый для вирусов и прочих онлайн-пакостей. Затем вы создаете транзакцию на компьютер, работающий онлайн, переносите ее на устройство USB, импортируете на компьютер оффлайн, подписываете ее, копируете подписанную транзакцию снова на устройство USB, импортируете ее на компьютер онлайн и передаете в сеть Биткойн.

Другое решение – *Bitcoin Wallet*, приложение для устройств Android. Однако не советуем хранить слишком много биткойнов в мобильном кошельке, потому что, потеряв смартфон, вы потеряете и все свои биткойны. Более того, перед удалением приложения, данных приложения или полной очисткой устройства от данных не забудьте перевести свой баланс на другой адрес Биткойн. При известных мерах предосторожности, *Bitcoin Wallet* будет удобным приложением для использования Биткойн без доступа к своему компьютеру.

Если вы хотите расплатиться с кем-то из своих друзей с помощью Биткойн, это будет очень просто, если у них обоих *Bitcoin Wallet* установлен на телефоне. Это приложение показывает код QR рядом с адресом Биткойн, и если ваш друг нажмет на код QR, он увеличится до размеров экрана. Затем, в *Bitcoin Wallet* нажмите на стрелку вправо в верхней части приложения, а затем нажмите на QR. Потом наведите камеру своего телефона на код QR адреса вашего друга, после чего ваше приложение использует этот адрес в форме. Выберите сумму платежа и опциональный взнос и нажмите на Send. Если все прошло нормально, ваш друг сразу же увидит, как зачисляется платеж. LXF

Скорая помощь

Клиент *Electrum* генерирует кодовую фразу, состоящую из 12 случайно выбранных коротких слов, которые используются для создания приватных ключей ваших адресов Биткойн. И пока вы помните кодовую фразу, вы всегда сможете восстановить свой баланс Биткойн, даже если потеряете свой кошелек.

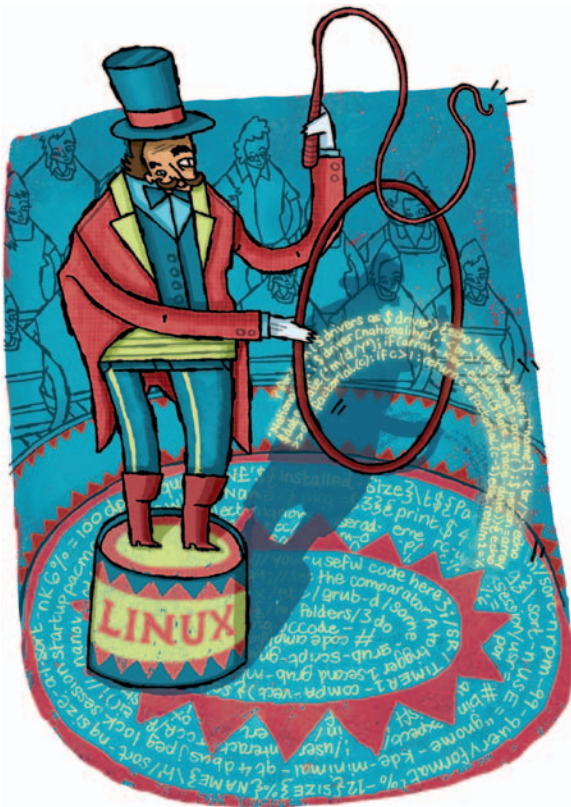
Django: Первый

Джоно Бэкон объясняет, как начать работу, и приглашает нас проехаться на Ferrari фреймворков для web-приложений... *Django*.



Наш эксперт

Джоно Бэкон – менеджер сообщества Ubuntu, автор книги «Искусство сообщества» и основатель ежегодного саммита руководителей сообществ.



Интернет в самых разных формах существует уже давно. На заре его развития содержимое было в основном статическим и текстовым, а с годами на наших глазах стало более динамичным и интерактивным и наполнялось видеороликами, звуком, анимацией – все это благодаря надежным браузерам.

Известных социальных сетей, сайтов для загрузки аудио- и видеоконтента и других отвлекающих от праведных трудов источников ныне хватает, но иной раз хочется создать собственный полноценный сайт с нуля и разместить его в сети. Это может быть ваша домашняя страница, система управления библиотекой, база данных музыки, система управления записями компании или лю-

бая из миллионов других разновидностей сайтов. Разумеется, при желании можно создать почти любой сайт, но с чего начать?

К счастью, существует масса web-фреймворков, позволяющих легко настроить и запустить сайт. Часть из них – готовые системы управления контентом, настраиваемые в соответствии с вашими потребностями, например, WordPress; но есть и более сложные платформы для развертывания сайтов. Эти обладают гораздо большей гибкостью, и одна из лучших среди них – *Django*.

Django – фреймворк с впечатляющим набором возможностей, который позволяет создавать web-приложения легко и быстро. Одна из главных причин популярности Django в том, что он основан на Python и обильно использует его. Тем из вас, на кого еще не снизошло просветление могущества и известности Python, скажем, что это язык программирования высокого уровня, который делает программирование простым, гибким и мощным. Сам Python – на удивление приятная и веселая штука, а в сочетании с *Django* – это средство, с помощью которого можно быстро создавать web-приложения и пользоваться изобилием встроенных функций Python. На нашем уроке мы пройдемся по созданию первого приложения с *Django*. Предполагается, что у вас есть некоторые познания в Python, но лишь самые базовые. К концу урока у вас должны появиться солидная основа, на которой можно будет наращивать и расширять ваши умения в *Django*.

Начинаем

Установку *Django* я подробно описывать не буду; в вашем дистрибутиве должны быть соответствующие пакеты (например, в Ubuntu можно установить *python-django*). Если у вас его нет и вы не хотите возиться с довольно витиеватым набором зависимостей, загрузите отличный стек *Bitnami*, с уже настроенным *Django*, с сайта <http://bitnami.org/stack/djangostack>.

Первым делом нужно открыть терминал. В нем мы будем выполнять команды для создания проектов и приложений, обновления базы данных и запуска временного web-сервера. Начнем, впрочем, с создания нового проекта. Для этого выполните команду

```
django-admin.py startproject pawnstore
```

Сознаюсь честно: мне безумно нравится реалити-шоу “Pawn Stars [Звезды заклада]” (оно рассказывает о трудностях и испытаниях одного ломбарда в Лас-Вегасе). Вот мы и напишем web-приложение для управления ломбардом. Здесь командой `django-admin` мы создали новый проект ‘pawnstore’ [ломбард].

В *Django* внутри одного проекта можно создать несколько приложений. Например, у вас может быть одно приложение для описания содержимого ломбарда, одно для управления продажами и одно для управления сменами. Все они могут быть в одном проекте.

Прежде чем создавать наше первое приложение, следует немного настроить проект. Внутри созданного проекта ‘pawnstore’ вы видите подкаталог `pawnstore`. В нем есть файл `settings.py` с параметрами всего проекта. Откройте его, и давайте сначала выберем базу данных, которой мы хотим пользоваться.

» Рис. 1. В интерфейсе администратора Django удобно добавлять, просматривать и удалять данные.



Ваш web-опыт

Django поддерживает множество различных СУБД (в том числе *SQLite*, *MySQL*, *PostgreSQL* и *Oracle*). Простоты ради, воспользуемся *SQLite*; эта простая база данных хранится в файле на вашем компьютере. Для этого найдите строку 'ENGINE' и добавьте в нее **sqlite3**. Она должна выглядеть так:

```
'ENGINE': 'django.db.backends.sqlite3',
```

Также нужно задать имя базы данных. Добавьте следующую строку:

```
'NAME': 'pawnstore.db',
```

Введя все настройки, можно создать первое приложение в проекте. Мы создадим приложение *Django* для управления списком товаров ломбарда и назовем его 'inventory'. Для создания приложения мы сперва воспользуемся скриптом **manage.py** (в каталоге верхнего уровня **pawnstore**). Этот скрипт – наш швейцарский армейский нож в управлении проектом. В дальнейшем мы будем пользоваться им все чаще и чаще.

Чтобы создать приложение, выполните команду

```
python manage.py startapp inventory
```

Она создаст новый каталог **inventory**. Теперь приступим к созданию приложения.

Создаем базу данных

Как и другие базы данных, *Django* хранит данные в таблицах и в полях этих таблиц. Нам будет нужно создать таблицы и поля и затем добавить данные, которые можно будет запрашивать и отображать разными способами. К счастью, вместо других систем, в которых для создания таблиц, полей и добавления данных нужно создавать десятки SQL-запросов, мы воспользуемся для этого славной своей мощью и простотой Python.

Внутри каталога **inventory** есть файл **models.py**. В этот файл можно добавить классы Python с описанием структуры данных нашего приложения для управления перечнем товаров. Мы попросту создадим две таблицы: **Categories** [категории] и **Items** [товары]. Каждый товар относится к какой-либо категории.

Чтобы добавить упомянутые таблицы, впишите следующий код в **models.py**:

```
class Category(models.Model):
    name = models.CharField(max_length=200)
class Item(models.Model):
    category = models.ForeignKey(Category)
    dateadded = models.DateTimeField('date added')
    name = models.CharField(max_length=200)
    description = models.TextField()
    quantity = models.IntegerField()
```

Здесь каждый класс соответствует своей таблице в нашей программе, а каждая переменная класса с помощью функции *Django* описывает тип поля с данными. Например, **dateadded** в классе **Item** имеет тип **DateTimeField**, поэтому *Django* будет хранить данные поля в формате даты, а в некоторых случаях, когда пользо-

Узнайте больше

Если вы заинтересовались *Django* и хотите узнать о нем подробнее, обязательно зайдите на замечательный сайт проекта и почитайте фантастическую документацию. Мир Django доступен на www.djangoproject.com.

вателю нужно ввести дату, выводить календарь для выбора даты. Другой пример – поле 'name', его тип **CharField**; это текстовая строка, и данные этого поля обычно вводятся в однострочное текстовое поле, в отличие от **TextField**, с которым обычно используется многострочное текстовое поле.

Схема нашей базы данных создана; теперь с ее помощью сгенерируем базу данных. Для этого скопируйте

```
python manage.py syncdb
```

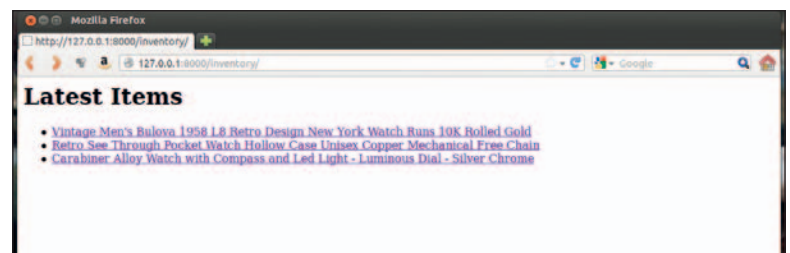
Тут вам зададут несколько вопросов. На вопрос, хотите ли вы создать суперпользователя, ответьте «да» и задайте имя пользователя и пароль. Затем вы должны вернуться в командную строку. База данных создана успешно. Запустим наш проект и посмотрим, как он выглядит. Для удобного просмотра проекта в *Django* есть встроенный web-сервер. Запустит его команда

```
python manage.py runserver
```

Запустите браузер и откройте адрес <http://127.0.0.1:8000/>, чтобы просмотреть проект. Вы должны увидеть текст о «первом приложении на Django», выполненный стандартным шрифтом. Хотя это не слишком впечатляет, но сдвиг есть! Кстати, остановить сервер можно, нажав Ctrl+C.

Интерфейс администратора

Наша база данных создана, и пора добавить в нее немного данных. В других фреймворках нам пришлось бы опуститься до написания целых кип сложных SQL-запросов, где легко наделать ошибок, или воспользоваться сторонним клиентом SQL, например, *phpmyadmin*. К счастью, с *Django* нам это не понадобится: в нем есть собственный интерфейс для управления базой данных, »



» Рис. 2. Мы здесь выводим простой список, но можно пользоваться любыми возможностями CSS.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Не повторяй себя

За созданием *Django* стоит множество принципов. Создатели проекта приложили множество усилий для максимального воплощения принципа DRY («Не повторяй себя» – Don't Repeat Yourself). Общая его идея в том, что в коде не нужно повторяться, а если вы повторяетесь, значит,

делаете что-то неправильно. Это особенно справедливо для *Django*, разработчики которого приложили массу усилий, чтобы уменьшить такие повторения. Всегда старайтесь пользоваться этим принципом в своих проектах; определяйте код один раз и используйте его многократно.

добавлением и удалением данных и других действий. Он встроен в ваш проект.

Чтобы включить интерфейс администратора, нужно задать кое-какие настройки. Во-первых, нужно настроить URL. Обсудим URL'ы в *Django* в общем случае. В каждом проекте *Django* есть файл **urls.py**. Его можно найти в подкаталоге проекта (у нас – **pawnstore**). Этот файл содержит список URL-адресов, распознаваемых проектом, в виде регулярных выражений. Адреса можно затем связать с различными приложениями проекта. Подробнее об этом позже.

В файле должна присутствовать следующая строка, которую нужно раскомментировать:

```
url(r'^admin/', include(admin.site.urls)),
```

Заодно раскомментируйте такие строки в начале файла:

```
from django.contrib import admin
admin.autodiscover()
```

Так мы включим интерфейс администратора в нашем проекте; но чтобы он поддерживался, нужно еще одно небольшое изменение. Откройте файл **settings.py** и раскомментируйте следующую строку в разделе **INSTALLED_APPS**:

```
'django.contrib.admin',
'inventory',
```

Раздел **INSTALLED_APPS** в **settings.py** описывает различные приложения в проекте, готовые для использования; одно из них – интерфейс администратора, другое – наше основное приложение 'inventory'.

Опять запустим сервер:

```
python manage.py runserver
```

Затем откройте адрес <http://127.0.0.1:8000/admin/>, и должна открыться форма для входа в интерфейс администратора. Введите имя пользователя и пароль – и появится главный интерфейс.

К сожалению, таблиц, которые мы создали ранее (Categories и Items) здесь нет – только Groups [группы], Users [пользователи] и Sites [сайты]. Причина в том, что хотя мы включили интерфейс администратора, мы еще не предоставили интерфейсу права доступа для редактирования наших таблиц. Для этого в каталоге **inventory** нужно создать файл **admin.py** и добавить в него следующие строки:

```
from django.contrib import admin
from inventory.models import Category, Item
admin.site.register(Category)
admin.site.register(Item)
```

Затем перезапустите сервер и перезагрузите интерфейс администратора. В нем появятся две новых таблицы, как на рис. 1. Воспользуемся этим и добавим немного данных. Во-первых, добавим какую-нибудь категорию. Щелкните по таблице Categories (заметьте, что к названию каждой таблицы добавляется окончание 's' для множественного числа... как мило!) и затем на кнопку Add Category [Добавить категорию] в правом верхнем углу. После этого укажите название категории и нажмите кнопку Save [Сохранить].

Теперь добавим товар [Item]. Вернитесь на домашнюю страницу и щелкните по таблице Items. Нажмите кнопку Add Item [Добавить товар] и обратите внимание, что формы для полей различных типов, которые мы добавили в модель базы данных, выглядят по-разному (отчасти поэтому важно задавать тип поля правильно). С помощью этой формы можно добавить товар в базу данных.

Одна из проблем, которую вы, возможно, заметили, состоит в том, что в выпадающем списке Category [Категория] не отображается ранее добавленная категория. Вместе этого мы видим там общий объект 'category'. Это легко исправить.

Причина, по которой это происходит, в том, что мы не задали поле по умолчанию, которое возвращается при обращении к классу в целом. Нам нужно сказать, что когда запрашивается объект Category, нужно вернуть заданное поле, если не запрашивается какое-то другое поле. Сделаем этим полем 'name', добавив следующую строку после класса Category в **models.py**:

```
def __unicode__(self):
    return self.name
```

А теперь раззадоримся и добавим в нашу базу побольше данных. В каждой категории должно быть не меньше трех товаров. Например, я добавил категорию 'Watches [Часы]' и скопировал и вставил несколько описаний с eBay.

Создаем виды

Наш следующий шаг – получить данные из базы и отобразить их на web-странице. Для этого мы создадим несколько способов (видов, views) для просмотра наших данных.

Для начала следует изменить **urls.py**, добавив в него URL для каждой страницы. Начнем с общего списка последних товаров, которые были добавлены в перечень. В **urls.py** добавьте следующий код:

```
url(r'^inventory/$', 'inventory.views.index'),
```

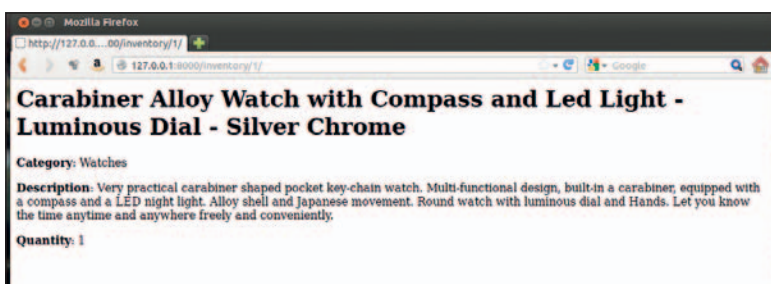
Эта строка сообщает *Django*, что если URL заканчивается на 'inventory' (например, <http://127.0.0.1:8000/inventory/>), то нужно загрузить для него страницу **inventory.views.index**. Что такое **inventory.views.index**? Так и думал, что спросите! Сейчас объясню...

При загрузке страницы Django заглядывает в **urls.py** и смотрит, нет ли там строки, которой соответствует URL. Если есть, загружается соответствующий вид. Вид – это просто функция, которая генерирует данные, необходимые для отображения содержимого страницы. Все виды описаны в **views.py** (несколько видов в одном файле).

Теперь создадим вид. Откройте **views.py** и добавьте следующий код:

```
from inventory.models import Item
from django.shortcuts import render_to_response
def index(request):
    latest_item_list = Item.objects.all().order_by('-dateadded')[:5]
```

➤ **Рис. 3. Динамические «движки» вроде Django предоставляют обширные возможности отображения и связывания данных.**



➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

```
return render_to_response('inventory/index.html', {'latest_item_list': latest_item_list})
```

В приведенном фрагменте кода мы создаем функцию `index()`. Имя этой функции ссылается на `index` в `inventory.views.index` из нашего файла `urls.py`. В следующей строке мы получаем все товары из таблицы `Items` и сортируем их в обратном порядке (это делает вызов перед `-dateadded`), а затем помещаем их в переменную `latest_item_list`. В следующей строке мы вызываем функцию `Django render_to_response()`, которая отправляет их шаблону `inventory/index.html`. Давайте же теперь разберемся, что это такое – шаблоны.

Шаблоны

Одна из традиционных проблем с web-приложениями состоит в том, что программерам нужно писать код, а дизайнерам – иметь возможность создать красивый дизайн, который отображает содержимое, сгенерированное кодом. Это пересечение навыков программирования и дизайна часто приносит с собой трудности; код может сбивать с толку дизайнеров, а программисты не всегда знают, как создать красивый дизайн. *Django* решает эту проблему с помощью шаблонов.

Пока мы задали тип URL и написали фрагмент кода, генерирующий информацию, которую нужно отобразить для URL этого типа. Последнее, что мы добавили – шаблон, который будет отображать этот код. Шаблон – специальный файл, который можно отдать дизайнеру, чтобы им было проще создать красивый дизайн на основе содержимого, генерируемого кодом, не загружая их самим кодом.

Чтобы приступить к работе с шаблонами, создадим каталог `templates` в нашем проекте. В нем создайте подкаталог `inventory`, а там – файл `index.html`. Прежде чем создавать шаблон, нужно сказать *Django*, где находятся шаблоны. Добавьте в файл `settings.py` полный и абсолютный путь к ним в разделе `TEMPLATE_DIRS`. Например:

```
TEMPLATE_DIRS = (
    "/home/jono/pawnstore/templates",
)
```

Внутри каталога `templates/inventory` должен быть файл `index.html`. Добавьте в него такой код:

```
<h1>Latest Items</h1>
{% if latest_item_list %}
<ul>
  {% for item in latest_item_list %}
    <li><a href="/inventory/{{ item.id }}">{{ item.name }}</a></li>
  {% endfor %}
</ul>
{% else %}
<p>No items are available.</p>
{% endif %}
```

В этом файле используется особая разметка, которую *Django* преобразует в обычный HTML при загрузке шаблона. Обратившись к шаблону в `views.py`, мы передали ему данные в `latest_item_list`. Затем используются обычные операторы: `if`, `for` и `else` в блоках `{ % }` для отображения данных в соответствии с теми или иными условиями. Отметим, что мы создали ссылку, к которой до-



➤ На сайте *Django* есть первоклассная документация по всем компонентам фреймворка.

бавляется идентификатор товара (например, `http://127.0.0.1:8000/inventory/3/`) – такой формат URL можно использовать для отображения подробной информации о товаре при щелчке на нем. Наш шаблон в действии можно увидеть на рис. 2.

Для начала откроем `urls.py` и добавим строку для загрузки новой страницы на тот случай, когда в конце адреса передается идентификатор товара (например, `http://127.0.0.1:8000/inventory/3/`):

```
url(r'^inventory/(?P<item_id>\d+)/$', 'inventory.views.detail'),
```

Здесь формат URL задается регулярным выражением, и если адрес соответствует ему, мы загружаем вид `'detail'`. Важно отметить, что числу в конце адреса в этой строке соответствует переменная `item_id`. Мы ссылаемся на нее на этой странице.

Теперь добавим следующую функцию в наш файл `views.py`:

```
def detail(request, item_id):
    item = Item.objects.get(pk=item_id)
    return render_to_response('inventory/detail.html', {'item':item})
```

Здесь вы видите, как переменная `item_id` передается функции, в которой с помощью функции `Django get()` проверяется соответствие этой переменной первичному ключу товара (`pk`). Затем для передачи остального шаблона мы вызываем все ту же функцию `render_to_response()`.

Теперь создайте файл `detail.html` в каталоге `templates/inventory` и добавьте в него следующий код:

```
{% if item %}
<h1>{{ item.name }}</h1>
<p><strong>Category</strong>: {{ item.category }}</p>
<p><strong>Description</strong>: {{ item.description }}</p>
<p><strong>Quantity</strong>: {{ item.quantity }}</p>
{% else %}
<p>No items are available.</p>
{% endif %}
```

Здесь мы просто выводим все поля. Теперь можно щелкнуть на элементе списка и получить подробную информацию о нем, как показано на рис. 3.

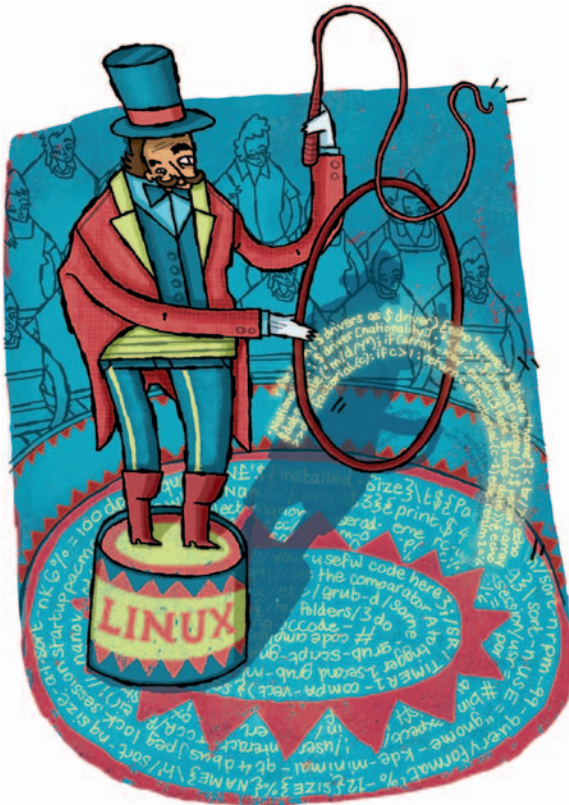
На нашем уроке мы познакомились со многими ключевыми идеями создания приложения в *Django*. Мы продвинулись далеко вперед, но это только начало. Нашей целью было познакомить вас с основными принципами, на основе которых вы сможете двигаться дальше. С фантастическим набором инструментов *Django* вы сможете создавать любые приложения, какие вам только понадобятся. Наслаждайтесь! **LXF**

Обработка форм

Уверен, что многие из вас захотят пойти дальше и узнать, как выводить формы и обрабатывать данные, введенные пользователем в форму. Более подробную информацию об этом можно найти по ссылке <https://docs.djangoproject.com/en/dev/topics/forms>.

Geany: Пишем

Спустя годы поисков идеального редактора для программирования, Джоно Бэкон уверовал, что, возможно, наконец-то он набрел на золотую жилу...



Написание кода – вещь глубоко личная. У каждого есть свои любимые инструменты, технологии и подходы к тому, как же написать превосходный код, который обеспечит выдающуюся функциональность. Сотни книг были написаны о разных аспектах Дзен написания кода, и я здесь явно не для того, чтобы поведать вам о том, что именно мой вариант является самым лучшим.

Ключевым моментом в этих спорах является то, нужно ли использовать в качестве окна в ваш код IDE или редактор. Любопытно, что многие сторонники IDE частенько являются начинающими или не очень опытными программистами, которые считают, что именно IDE крайне важны для снижения планки написания приложений; а многие умудренные жизнью хаекеры предпочитают редактор, дабы избежать всякой ненужной ерунды и сразу приступить к делу. Если говорить о моих предпочтениях, я вижу и сильные, и слабые стороны в обоих лагерях; мне всегда казалось, что это – дело исключительно личного выбора, и никто здесь не может быть прав полностью, кроме меня, разумеется.

За долгие годы я испробовал разные IDE: Eclipse, KDevelop, а совсем давно – Visual Studio. Я пробовал также и редакторы: Emacs, Vim и GEdit. На сегодняшний день я сделал свой выбор, выбор ясный и уверенный, и этот выбор – Geany.

Простота и мощь

Geany, по сути, текстовый редактор, приукрашенный и усиленный всякими штуками, потребными большинству программистов. На самом деле Geany не является ни IDE, ни обычным текстовым редактором; он являет собой текстовый редактор с важными вставками из IDE. Geany не пытается вместить в себе все эксперт-программы, шаблоны для начинающих и прочие моменты, обычные для IDE, но зато он включает основные функции, необходимые для написания приложений – например, отличный редактор, навигатор классов, свертывание кода и прочее. Я твердо убежден, что Geany является лучшим выбором для профессионального программиста или для любителя, пишущего приложения для Linux.

Geany написан с помощью набора виджетов GTK, а значит, отлично интегрируется в рабочие столы Unity и Gnome в Ubuntu и Fedora соответственно. Пакеты доступны через Ubuntu Software Centre, в репозитории Fedora Everything, а также для Debian, Gentoo, OpenSUSE, ArchLinux, Centos, RHEL, AltLinux, SourceMage и Crux, в www.geany.org/Download/ThirdPartyPackages.

Имеется встроенная поддержка обширного ряда языков программирования, в том числе (O)Caml, Ada, Assembler, C, C#, C++, COBOL, Python, D, Fortran, FreeBasic, Genie, GLSL, Haskell, Haxe, Java, Objective C, Pascal, Scala, Vala, VHDL и Verilog. Он также включает отличную поддержку языков скриптов и разметки, таких, как ActionScript, CMake, Erlang, Ferite, Forth, Javascript, LISP, Lua, Makefile, Matlab, NSIS, Perl, PHP, Python, R, Ruby, Shell, Tcl, CSS, DocBook, HTML, LaTeX, Markdown, reStructuredText, Txt2Tags и XML. В качестве бонуса, он обеспечивает отличную поддержку файлов Asc, Config, Diff, Gettext, SQL, и YAML. В Geany можно использовать и другие языки, просто им могут не подойти встроенное выделение синтаксиса и другие функции.

На данном уроке мы пройдемся по Geany и познакомимся со многими его функциями. Чтобы проиллюстрировать все это, я создам новый проект, создам пример программы, и затем мы пройдемся по разным функциям и опциям настройки, чтобы подогнать ваш опыт программирования под ваши нужды.

Пример кода я буду писать на Python, но вы можете использовать любой язык по своему усмотрению. Итак, начнем.

Интерфейс

Загрузите Geany, и вы увидите интерфейс, напоминающий изображенный на рис. 1. В этом интерфейсе имеются три основные части. Слева находится браузер. Здесь вы получаете быстрый доступ к разным файлам, классам, функциям и прочим символам, являющимся частью вашего проекта. Мы рассмотрим его чуть позднее, когда создадим наш код.

Справа находится основной текстовый редактор. Именно здесь вы будете проводить большую часть своего времени. Так же, как и во многих современных текстовых редакторах, вы можете загружать многочисленные файлы, и они доступны в многочисленных вкладах.

Под текстовым редактором находится раздел Messages [Сообщения]. Здесь вы увидите информацию от вашего компилятора/

КОД

интерпретатора об отладке программы, и другие полезные функции, например, место для написания заметок и просмотра задач, которые еще предстоит выполнить в вашем проекте.

В верхней части экрана вы видите меню (в Ubuntu меню встроено в верхнюю панель, как и большинство приложений) и обычные функции в приложении, доступные в виде панели инструментов. Панели инструментов также поддаются индивидуальной настройке. Чтобы настроить их, щелкните правой кнопкой по панели инструментов и выберите *Toolbar Preferences* [Параметры панели инструментов]. Теперь щелкните по кнопке *Customize Toolbar* [Настройка панели инструментов], чтобы вывести редактора панели инструментов. Затем просто выберите функции слева и щелкните по правой кнопке со стрелкой, чтобы добавить их к панели инструментов.

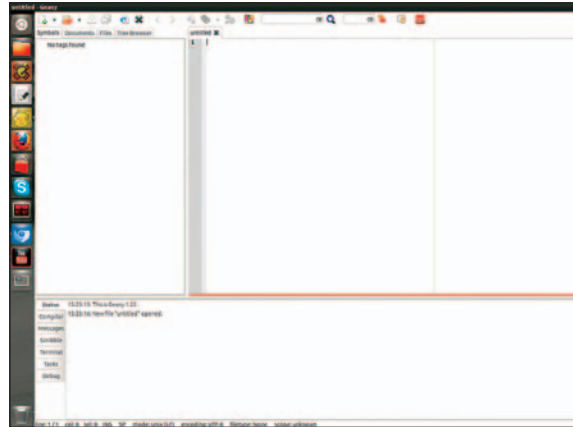
Пишем код

Давайте начнем – и создадим новый проект. Щелкните по *Project > New* [Проект > Новый], и в появившемся окне введите имя для своего проекта (например, “Awesome App” – «Потрясающее Приложение»), затем щелкните по кнопке *Create* [Создать].

Итак, проект создан; давайте настроим для него предпочтения. Для этого щелкните по *Project > Properties* [Проект > Свойства]. Весьма удобно начинать настраивать проект со вкладки *Indentation* [Отступы]. Здесь вы можете настроить, будут ли использоваться табуляция/пробелы. Это особенно важно для разработчиков Python, у которых могут быть проекты с применением табуляции или пробелов.

Если вы работаете над проектом, которому нужен компилятор (например, над приложением C/C++), вы также можете использовать вкладку *Build* для настройки команд для создания проекта. Это так же просто, как ввести команду и рабочую директорию для вашего кода. Мы не будем это рассматривать, потому что сегодня займемся Python.

Вы, вероятно, заметили, что за время создания проекта у нас ни разу не появился ни всплывающий мастер, ни набор опций для создания и отображения вашего приложения. Помните, я сказал, что *Geany* не является настоящей IDE? Вот как раз поэтому. Если вы хотите сгенерировать свое приложение, вам понадобится



► Рис. 1. Редактор *Geany* отличается простотой и отсутствием излишеств.

другой инструмент, типа *Quickly*, и затем вы сможете скопировать сгенерированные файлы в свою директорию проекта *Geany*.

В нашем новом проекте создайте новый файл, нажав на *File > New (with Template) > main.py* [Файл > Новый (с шаблоном) > main.py]. Создастся новый файл Python, в который уже включен некий код, например, комментарий по его лицензированию, и шаблонный код, чтобы реализовать функцию Python `main()`. Теперь нажмите на *File > Save As* [Файл > Сохранить как] и сохраните файл как `app.py`. Теперь давайте добавим знаменитое «Привет, мир! [Hello World!]» в `main()`, чтобы это выводилось на экран при запуске:

```
print "Hello World!"
```

Сохраните файл, а затем давайте его запустим.

В *Geany* файл можно запустить, сначала щелкнув по вкладке *Terminal* [Терминал] в нижнем окне вывода. Затем надо перейти в директорию с кодом (в моем случае это `/home/jono/projects/Awesome App/`) и запустить файл:

```
cd projects/Awesome App
python app.py
```

Теперь на экране должно появиться «Привет, мир!». Теперь давайте пробежимся по тому, как *Geany* уже делает наш код проще для навигации.

Слева, в браузере, щелкните по вкладке *Symbols* [Символы]. Это вкладка, на которой вы будете проводить больше всего времени в поисках разных частей кода. Вы уже должны увидеть, что эта вкладка показывает нашу основную функцию `main()`. Щелкнув по вкладке *Documents*, вы также сможете увидеть наш единственный исходный файл; это особенно удобно, если у вас много исходных файлов.

Щелкните по функции `main()` во вкладке *Symbols*, и вы увидите, что на полях редактора появилась маленькая стрелка, ►

10 лучших функций Geany

Зачем вообще связываться с *Geany*? Вот вам на это 10 причин:

- 1 Простота в использовании** *Geany* отличается простотой и отлично вписывается во все проекты программирования.
- 2 Навигация по коду** С помощью браузера классов и файлов их поиск в паутине кода станет проще.
- 3 Поддержка языка и выделение синтаксиса** Поддержка большого числа языков, в том числе C, C++, Python, PHP, Perl, Ada, Java, Ruby, Tcl, HTML, CSS, и многих других.

4 Автозаполнение символических имен Не можете вспомнить имя нужной функции? *Geany* предложит вам варианты, а также аргументы.

5 Автозаполнение тэгов XML и HTML При написании HTML и XML заполнение тэгов помогает сохранить разметку и облегчает процесс написания.

6 Сворачивание кода Скрывает те блоки кода, на которые вам нет нужды смотреть, через простое встроенное управление сворачиванием кода.

7 Модули расширения *Geany* предлагает широкий выбор сторонних модулей, которые можно

установить, чтобы разными способами обогатить редактор.

8 Поддержка задач Отображает все эти комментарии `TODO` и `FIXME` в одном месте, что облегчает выполнение и отладку соответствующих задач.

9 Встроенный терминал Встроенный терминал проводит тестирование частей кода в интерпретаторе или находит файлы быстро и легко.

10 Поддержка отладчика Интегрированный отладчик облегчает поиск и отслеживание проблемных частей кода.

► Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

указывающая на функцию. В редакторе также имеется маленький белый квадрат с символом - внутри, рядом с каждым блоком кода. Этот квадратик являет собой простой вариант сворачивания кода, чтобы скрыть те его фрагменты, которые вам на данный момент не нужны. Если щелкнуть по этому квадратику, он будет показывать символ +, сообщая, что блок кода свернут.

Теперь давайте создадим класс, чтобы продемонстрировать, как *Geany* может с легкостью осуществлять навигацию по вашему коду. Добавьте в свою программу следующий код:

```
class MyClass:
def __init__(self):
print "Создание объекта MyClass"
def show_info(self):
print "Это класс MyClass"
def foo(self, value):
print "Вы снабдили меня %s " % величиной
if __name__ == '__main__':
obj = MyClass()
obj.show_info()
obj.foo(5)
```

В этом коде я создаю простой (и довольно бесполезный) класс. Заметьте, что мы удалили функцию `main()`, поскольку она больше не нужна.

По мере написания кода вы, возможно, заметили, что рядом с каждой функцией класса, которую вы добавляете, автоматически появляются маленькие квадратик сворачивания кода. *Geany* умен: он наблюдает за кодом, который вы пишете, и автоматически применяет эти функции, не дожидаясь, когда вы завершите блок или сохраните файл.

Вы должны были заметить еще одно свойство: когда вы вызываете функции класса (т.е. вводите **obj** и затем имя функции), *Geany* выводит маленькое окошко автозаполнения кода, показывающее вам функции, которые, возможно, подойдут к тому коду, который вы ввели. Это очень удобная опция, и *Geany* будет часто пытаться предложить имена функций в разных API, которыми вы тоже пользуетесь.

Добавив наш класс к нашей программе, загляните во вкладку Symbols в браузере в левой части окна. Теперь вы видите, что добавленный класс появился рядом с маленьким голубым значком, и функции класса отображены как часть иерархии этого класса. Для такого небольшого приложения, как это, отображение класса не дает каких-то особых преимуществ, но для крупного при-

ложения, вроде показанного на рис. 2, этот браузер классов является исключительно важным, поскольку с его помощью легко перемещаться по кодовой базе в поисках того, что вам нужно.

Настройка

Одним из ключевых факторов, отличающих IDE от редактора, является уровень настройки. Многие IDE предоставляют ряд разумных функций по умолчанию и ограниченную возможность настройки, чтобы упростить работу в IDE для пользователей. Хотя многие редакторы не имеют удобной функции интерфейса, предлагаемой IDE, уровень настройки они предлагают превосходный. И *Geany* является одним из таких примеров.

Geany предлагает фантастический уровень настройки для многих частей процесса программирования. Вы можете настроить *Geany* через две области:

1 **Edit > Preferences [Редактировать > Предпочтения]** – именно здесь вы настраиваете основное приложение.

2 **Edit > Plugin Preferences [Редактировать > Предпочтения модулей]** – здесь вы можете настроить модули расширения (мы скоро поговорим о модулях-плагинах более подробно).

Давайте рассмотрим некоторые части основной работы в *Geany* (без расширений), которые вы, возможно, решите настроить.

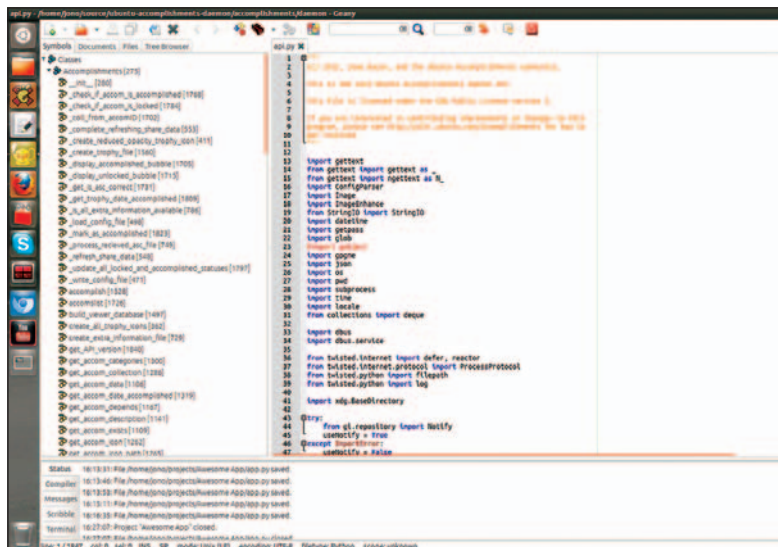
Начнем с настройки информации, добавляемой к комментариям заголовка при добавлении нового файла к проекту. Нажмите на Edit > Preferences > Templates [Редактировать > Предпочтения > Шаблоны], и вы увидите разные области, подлежащие редактированию. *Geany* постарается заполнить многие области за вас, но некоторые вам придется отредактировать самим.

В некоторых областях вы увидите нечто похожее на полную бессмыслицу, например, %Y-%m-%d в поле Date. Это коды, которые выступают в роли шаблонов, чтобы структурировать отображение некоторых вещей, например, даты. Например, %Y-%m-%d может конвертироваться в 2012-12-25 (%Y – это 2012, %m – 12, и %d – 25). Полный список этих кодов и того, что они делают, вы найдете на <http://man.cx/strftime>.

Идем дальше. Одной из самых важных областей настройки для многих разработчиков является привязка клавиш. Программирование – это во многом и наука, и искусство, в котором есть ритм и каденция. Когда вы «в ударе», вы стремитесь сохранить свой ритм, поскольку он отражается на производительности вашего написания кода. Именно по этой причине многие из нас نحن напролет сидят за программированием, не желая потерять кураж.

А если что и сбивает с куражу чаще, чем все остальное, то это необходимость использовать мышь. Отрыв рук от клавиатуры, чтобы перенести их на мышь, а потом обратно, сильно замедляет процесс, а функциональные клавиши дают вашему куражу возможность куражиться и дальше. Если раньше вы никогда не использовали функциональных клавиш, настоятельно вам рекомендую их изучить, и очень хорошо, что в *Geany* у вас есть возможность настроить практически все функциональные клавиши.

➤ Рис. 2. *Geany* предоставляет простые инструменты для управления крупными проектами.



Модули расширения

Одним из самых чудесных свойств *Geany* является широкий ряд сторонних модулей, которые можно установить. Многие из них уже доступны в соответствующем менеджере пакетов (например, в *Ubuntu Software Centre* или через *Yum*), но вы также можете найти более подробную информацию о модулях на <http://plugins.geany.org>.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

А контроль версий?

Одним из главных вопросов, который задают о *Geany*, является вопрос о наличии встроенной поддержки для Git, Bazaar, Subversion, CVS и других систем контроля версий. По умолчанию *Geany* не включает этой поддержки, но для нее есть специальный модуль расширения; более подробно об этом можно узнать на <http://plugins.geany.org/geanyvc.html>.

Для этого щелкните по Edit > Preferences и нажмите на вкладку Keybindings [Привязки клавиш].

Еще одна важная область настройки – сам редактор. Хотя текстовый редактор кажется весьма простым... вы печатаете в нем текст... все же есть множество факторов, в которых и кроются различия в использовании текстового редактора. Отступы... выделение синтаксиса... нумерация строк... перенос строки... новые строки и конечные пробелы... автозаполнение кода и прочие элементы – все они и являются этими факторами. К счастью, в *Geany* имеются опции настройки для широкого ряда функций в Edit > Preferences > Editor [Редактировать > Предпочтения > Редактор].

Вы также можете настроить то, какие части редактора будут видимыми, нажав на View > Editor [Вид > Редактор].

Я бы порекомендовал вам обязательно включить Line Numbers [Номера строк]; возможность увидеть номер обсуждаемой строки важна при сотрудничестве с другими разработчиками.

Можно также использовать View > Editor > Colour Schemes [Вид > Редактор > Цветовые схемы], чтобы загрузить альтернативную цветовую схему редактора, если, например, вы предпочитаете более темные тона. Сменные темы можно найти на www.geany.org/Download/Extras#colors.

Обработка задач

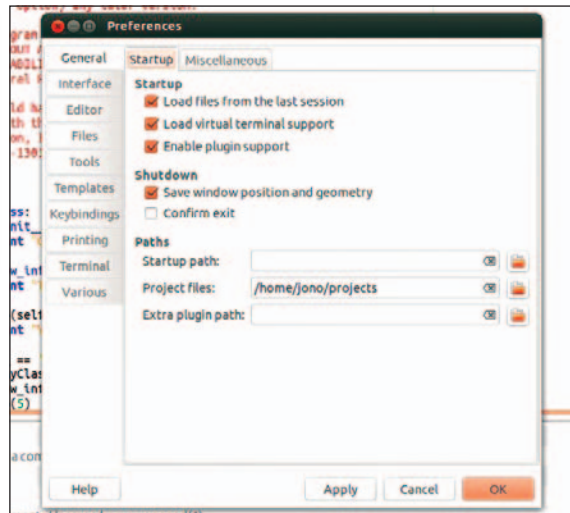
В программировании обычным делом является добавление к коду комментариев, указывающих на то, что надо исправить в определенных частях кода. Например, если какая-то функция недоделана, вы можете добавить к ней следующее, чтобы ее выделить:

```
# TODO: Expand this to include support for XYZ. [Расширить, чтобы добавить поддержку XYZ]
```

Вы также можете оставить комментарий, чтобы исправить часть кода позднее. Частенько это выглядит так:

```
# FIXME: Make this thread-safe. [Сделать поточно-ориентированным]
```

Одна из проблем с подобным подходом в том, что может оказаться трудным обнаружить эти небольшие примечания позднее, чтобы успеть решить проблемы до выхода релиза.



► Рис. 3. Geany идет с отличным набором разумных настроек по умолчанию.

К счастью, *Geany* включает небольшую функцию, применяемую для отображения всех этих примечаний в одном месте, создав таким образом список неотложных дел (TODO, Надо сделать) для своего кода. Чтобы использовать эту функцию, просто используйте TODO и FIXME [поправь меня] для описания проблемы, как показано выше, и убедитесь, что в Edit > Plugin Preferences > Addons [Редактировать > Предпочтения модулей > Дополнения] у вас отмечена функция Show available tasks [Показать имеющиеся задачи] в окне Messages. Если захотите, можете установить другие термины вместо TODO и FIXME, но TODO и FIXME являются повсеместно используемыми, так что вы, возможно, решите просто придерживаться настроек по умолчанию.

Geany – простой, но невероятно мощный редактор для программирования, который дает вам все основные инструменты, необходимые для написания кода. Разработчики *Geany* сосредоточили все свое внимание на простоте и настоящих потребностях программистов, не шпигуя его всякими красотами, которые большинству программистов совершенно не нужны.

Это придает редактору истинную зрелость, и, к счастью, уровень настройки и богатый выбор модулей расширения означает, что вы действительно можете настроить *Geany* под свои потребности.

Изучение *Geany* не займет у вас много времени, и эта статья продемонстрировала вам главное, что вы должны знать, чтобы работать продуктивно.

Итак, не трата времени на всякую ерунду, приступайте-ка к программированию. Счастливого вам программирования, друзья мои! LXF

Альтернативы Geany

Хотите организовать здоровую конкуренцию? Вот вам несколько альтернатив *Geany*:

► **Eclipse** появился на сцене IDE уже давно, и предоставляет многоплатформенную, многоязычную IDE, разработанную с учетом потребностей разработки приложений уровня предприятия. Хотя она и отличается некоторой сложностью, а временами и громоздкостью, все же Eclipse являет собой фантастическую IDE с отличной поддержкой множества языков.

► **KDevelop** изначально разрабатывался для разработчиков KDE и с учетом их потребностей; проект KDevelop был одной из первых IDE, созданных

специально для среды рабочего стола. Предлагающий всесторонний набор функций KDevelop является отличным выбором, особенно если вы собираетесь писать приложения для рабочего стола KDE.

► **GEdit** На поверхности *GEdit* выглядит простым текстовым редактором, но в нем есть разнообразные плагины и расширения, которые могут превратить *GEdit* в нечто подобное *Geany*. Если при программировании вы предпочитаете простые редакторы, то *GEdit* – отличная опция.

► **Qt Creator** Созданный специально для создания приложений Qt и QML, Qt Creator является IDE

не просто для написания кода, но и для создания интерфейсов, просмотра документов и прочих функций. Тем, кому надо написать приложение Qt, настоятельно его рекомендуем

► **Anjuta** Уходя корнями в долгую историю разработки, Anjuta изначально предназначалась как IDE для написания приложений Gnome, но эта многогранная IDE также предлагает поддержку и для других видов приложений.

► **Bluefish** Если вы хотите писать только web-контент, такой, как HTML, PHP, или Python в web-контексте, то *Bluefish* будет совершенно фантастическим web-редактором.



Udev: Отшлифуйте

Есть место, в котором Linux прячет свои самые глубокие и темные секреты. Нейл Ботвик заглядывает туда. Снова.



Наш эксперт

У **Нейла Ботвика** по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не кажет, где находится центральный сервер.

Кредо Unix «Все является файлом» относится и к устройствам. Когда вы вставляете флэшку или подключаете web-камеру, в каталоге `/dev` появляется узел устройства (файл), но как он туда попадает? В старые добрые времена нужно было прибегать к тайным заклинаниям `mknod` и обращаться к священному списку старших и младших номеров устройств; теперь подключить устройство достаточно просто. Новая волшебная команда – `udev` – работает как динамическая файловая система и как менеджер устройств. При загрузке системы `udev` создает в каталоге `/dev` виртуальную файловую систему вроде `/proc` или `/sys`, с файлами устройств для всей периферии, обнаруженной ядром. Затем она слушает события горячего подключения, поступающие от ядра, и на лету создает или удаляет файлы устройств для накопителей, сетевых адаптеров, устройств для работы с видео и всего, что можно подключить «на лету», когда эти устройства подключаются или отключаются от компьютера.

На первый взгляд она ничем не отличается от своей предшественницы `devfs`, но `udev` работает полностью в пространстве пользователя, а не как процесс ядра. Ядро обнаруживает устройство, но не создает файл устройства, передавая ответственность `udev`. Это означает, что `udev` может выполнять какие-то действия до создания файла устройства.

Все это работает под управлением правил – текстовых файлов, где описаны условия и действия: если устройство соответствует условиям, то действия выполняются. Правила находятся в двух каталогах: `/lib/udev/rules.d` и `/etc/udev/rules.d`. Первый содержит правила, установленные `udev` и другими пакетами – например, с SANE устанавливается огромный файл правил, где определяются действия, выполняемые при обнаружении различных сканеров. Файлы в `/etc/udev/rules.d` содержат правила, устанавливаемые другими пакетами, и пользовательские правила – именно они нас интересуют. Прежде чем начинать эксперименты с фай-

лами в этом каталоге, учтите, что все файлы в нем уже установлены какими-то пакетами и могут быть перезаписаны при обновлении пакета, поэтому помещайте все свои правила в отдельный файл. Имена файлов имеют значение; они должны заканчиваться на `.rules` и считываться в определенном порядке, поэтому начинаются с числа. В расчет берется только имя файла, а не путь, так что файлы

```
/lib/udev/rules.d/10-foo.rules
/etc/udev/rules.d/20-bar.rules
/lib/udev/rules.d/30-stuff.rules
```

будут прочитаны именно в этом порядке. При наличии двух файлов с одинаковым именем используется только файл из `/etc`. Начните имя файла с 10, если правила должны выполняться первыми, или с 90, если последними. Порядок имеет значение, так как более поздние правила перегружают более ранние.

Условия и присваивания

Каждое правило – одна строка в файле, которая содержит несколько пар «ключ–значение». На практике нужны как минимум две пары: условие и присваивание. Если все условия верны, выполняется присваивание. Вот пример:

```
DRIVERS=="sd", ATTRS{vendor}=="Google", SYMLINK="android"
```

Первая часть до запятой – условие; она сравнивает значения с помощью оператора `==` и проверяет, является ли устройство устройством хранения данных. Если это так, проверяется следующая часть `ATTRS{vendor}`, поле стандартного атрибута, и если оно равно «Google», значит, это телефон Android. Если второе условие верно, выполняется присваивание, которое настраивает символическую ссылку с имени устройства, назначенного ядром – скажем, с `/dev/sdc` на `/dev/android`. Прекрасно, скажете вы, но как узнать, на что именно делать проверку? `Udev` поможет нам и с этим, благодаря команде `udevadm`. Вставьте флэшку, отметьте имя появляющегося устройства, например, `/dev/sdb1`, затем выполните команду

```
udevadm info --name /dev/sdb1
```

Она выдаст полезную информацию об устройстве, из которой вы можете понять, что надо использовать для сравнения, но это не формат правила `udev`. Чтобы получить ее в нужном формате, скомандуйте

```
udevadm info --attribute-walk --name /dev/sdb1 | less
```

То, что мы пропускаем вывод команды через `less`, намекает, что команда выдает большой объем информации. Она выводит не только информацию об устройстве, но и проходит по дереву, показывая атрибуты каждого из его родителей, от `/dev/sdb` до USB-контроллера. Для проверки можно выбрать любое из них, но нас на самом деле интересуют только один или два верхних блока. В файле с правилом `udev` нам нужен формат вывода, поэтому можно вставить соответствующую строку в файл с правилом. Если нужно сравнивать с несколькими атрибутами, все они должны быть в одном и том же блоке. Чтобы правило сработало, все условия должны быть верны. Если достаточно, чтобы было верным только одно условие, придется добавить отдельное правило для каждого условия.

➤ Список атрибутов устройства от команды `udevadm`. Здесь можно увидеть атрибуты «модель [model]» и «производитель [vendor]», используемые для идентификации устройства.

```

looking at parent device '/devices/pci0000:00/0000:00:1c.5/0000:06:00.0/usb5/5-2/5-2:1.0/h5
KERNELS=="15:0:0:0"
SUBSYSTEMS=="scsi"
DRIVERS=="sd"
ATTRS{rev}=="0.00"
ATTRS{type}=="0"
ATTRS{scsi_level}=="3"
ATTRS{model}=="Flash Voyager"
ATTRS{state}=="running"
ATTRS{queue_type}=="none"
ATTRS{iodone_cnt}=="0x2c1"
ATTRS{iorequest_cnt}=="0x2c1"
ATTRS{timeout}=="30"
ATTRS{evt_media_change}=="0"
ATTRS{max_sectors}=="240"
ATTRS{ioerr_cnt}=="0x1"
ATTRS{queue_depth}=="1"
ATTRS{vendor}=="Corsair"
ATTRS{device_blocked}=="0"
ATTRS{iocounterbits}=="32"

looking at parent device '/devices/pci0000:00/0000:00:1c.5/0000:06:00.0/usb5/5-2/5-2:1.0/h5
KERNELS=="target15:0:0"
SUBSYSTEMS=="scsi"
DRIVERS=="

looking at parent device '/devices/pci0000:00/0000:00:1c.5/0000:06:00.0/usb5/5-2/5-2:1.0/h5
KERNELS=="host15"
-- MOST: *stdin* (38,1) 24%
Press 'Q' to quit, 'H' for help, and SPACE to scroll.

```

СВОЙ КОМПЬЮТЕР

Добавить символические ссылки на файлы устройств просто, но можно сделать гораздо больше. Например, заглянув в `/dev/disk`, вы увидите подкаталоги, такие как `by-id` и `by-partlabel`. Они содержат символические ссылки для определения всех дисков и разделов по их UUID и меткам файловых систем. Если у вас в `/etc/fstab` используются UUID, вы увидите, на какие реальные дисковые устройства они ссылаются. Эти ссылки автоматически создаются при загрузке системы при запуске `udev`. Другие типы присваиваний, которые можно добавить в правило – **OWNER**, **GROUP** и **MODE**, они изменяют владельца и права доступа к файлу устройства. Например, если у вас есть сканер, который не могут найти SANE и GIMP, но root видит его, обычно это ограничения прав доступа к файлу устройства. Изменить их можно таким правилом:

```
ATTRS{product}=="CanoScan",
ATTRS{manufacturer}=="Canon", GROUP=="scanner",
MODE=="0660", SYMLINK=="scanner"
```

Это правило, которое я добавил для сканера Canon, обнаружившего именно такое поведение. Правило делает группу устройств сканеров доступной на запись, а ее владельцем становится группа scanner, поэтому любой пользователь в этой группе может им пользоваться. В однопользовательской системе можно просто установить режим в 0666, но группа scanner – это общепринятый способ.

Переименовывать файлы устройств больше нельзя, но сетевые интерфейсы – можно. Если у вас несколько сетевых карт, например, если компьютер является маршрутизатором или брандмауэром, было бы желательно, чтобы им всегда назначался один и тот же адрес. Переключение интерфейсов на брандмауэре способно вызвать катастрофу. В условии `udev` можно использовать MAC-адрес карты, указанный после "ether" в выводе `ifconfig`; например, с его помощью можно создать такое правило:

```
SUBSYSTEM=="net", ATTR{address}=="e8:11:32:09:63:82",
NAME=="eth0"
SUBSYSTEM=="net", ATTR{address}=="50:e5:49:bd:1e:45",
NAME=="eth1"
```

Если вы когда-нибудь меняли сетевую карту, возможно, вам было интересно, почему новая карта появлялась в системе как `eth1`. Все дело в правиле `udev`, которое автоматически создается в `/etc/udev/rules.d/70-persistent-net.rules`, оно очень похоже на правила выше. Так как старый адрес был выделен `eth0`, новому адресу будет выделено первое доступное имя устройства. Поэтому если вы заменили карту на новую, либо удалите старое правило из файла, либо удалите файл, и он будет вновь создан для новой карты при следующей перезагрузке системы. Постоянные сетевые правила были удалены из `udev`, и если вы хотите изменить сетевые имена на самых новых дистрибутивах, понадобится создать правила вручную в таком же формате, при котором сначала проверяется **SUBSYSTEM**, чтобы убедиться, что это сетевая карта, а затем MAC-адрес.

Не меняйте правила

Не поддавайтесь соблазну изменять существующие файлы с правилами; они устанавливаются менеджером пакетов и могут быть заменены при обновлениях. Вместо этого создавайте собственные файлы с правилами в `/etc/udev/rules.d`. Если вы хотите изменить существующее правило, скопируйте его

в собственный файл и выполните изменения там. Если задать файлу больший номер, чем у существующего файла с правилом, у вашего правила будет преимущество. Также, пользуясь знаком `:=` вместо `=`, можно задавать приоритет присваиваний – в этом случае не только выполняется присваивание, но и гарантируется, что ни одному из последующих правил не удастся изменить его на свое.

Наряду с изменением файлов устройств в `/dev`, `udev` также может запускать программы. Программа `usb_modeswitch` для 3G-модемов использует правила `udev` для переключения устройства из режима накопителя в режим модема, и теперь программу больше не нужно запускать вручную. Помните, что во время работы программы `udev` блокируется, поэтому пользуйтесь им только для программ, завершающихся быстро. Если нужно запустить нечто долгосрочное, поместите команду в скрипт и вызовите его из правила с `&`, чтобы он сразу вернул управление. Например, в правиле

```
KERNEL=="sd[a-z]1", ATTRS{model}=="DMC-TZ30", RUN="/usr/local/bin/copyphotos.sh $devnode &"
```

используется параметр **KERNEL** для определения имени устройства и атрибут **model** для определения конкретной модели камеры. При подключении этой камеры правило запускает скрипт, передавая ему в качестве аргумента имя устройства. Текст скрипта выглядит так:

```
#!/bin/sh
pmount $1 camera
mv /media/camera/DCIM/100_PANA/*.JPG /home/nelz/photos/new
pumont /media/camera
```

Скрипт монтирует камеру, используя имя устройства, переданное `udev`, перемещает фотографии в домашний каталог и размонтирует камеру. Демон `udev` выполняется от имени суперпользователя-root, и все запущенные им команды тоже будут выполняться от имени root. Это следует учитывать, особенно для команд монтирования устройств, потому что обычно они должны быть доступны на запись только root. В этот скрипт можно добавить команду для изменения владельца после копирования.

Более подробную информацию можно найти на map-странице `udev`, заглянув в файлы правил в своей системе и в руководстве в Интернете. Удобное руководство есть по ссылке www.reactivated.net/udevrules.php. [LXF](#)

» В каталоге `/usr/lib/udev` есть много файлов – редактировать их нельзя, но можно перезагрузить их собственными файлами правил в `/etc/udev/rules.d`.

SSH: СВЯЗЬ БЕЗ

Нейл Ботвик показывает, как с помощью SSH два компьютера могут общаться друг с другом максимально удобно и безопасно.

Про SSH, или «защищенную оболочку [Secure Shell]», слышали многие из нас, но она умеет не только открывать сеанс терминала на другом компьютере. Тем, кто не знаком с SSH, будет интересно узнать, что с ее помощью можно открыть защищенный сеанс терминала через сеть. Весь трафик, включая обмен паролями, шифруется, что позволяет использовать незащищенные каналы. Обычно SSH запускается так:

```
ssh user@hostname
```

Команда подключится к удаленному хосту и запросит пароль пользователя, а затем откроет оболочку по умолчанию для этого пользователя. Часть адреса `user@` можно опустить, и тогда SSH воспользуется текущим именем пользователя, но придется указать пароль; при этом будут использованы настройки по умолчанию для этого пользователя на удаленном компьютере.

Использовать пароли для входа и неудобно, и ненадежно; и чем надежнее пароль, тем это менее удобно. SSH имеет и альтернативу – аутентификацию по ключам, позволяющую пользователю войти в систему на другом компьютере от своего имени без ввода пароля. Поэтому не пользуйтесь ею на устройствах, которые легко потерять или украсть, например, на смартфонах. Создать ключи на локальном компьютере можно командой:

```
ssh-keygen -t edcsa
```

Параметр `-t` задает тип ключа: `rsa`, `dsa` или `edcsa`. Последний используется по умолчанию в более новых версиях SSH, но не поддерживается в более старых релизах, поэтому вы можете выбрать `dsa`. Ответьте на вопросы, и команда создаст пару ключей в каталоге `~/.ssh`; файлы будут названы в соответствии с типом, например, `id_edcsa` и `id_edcsa.pub`. Последний файл – публичный ключ, который копируется на удаленный компьютер, другой файл – приватный ключ, и им не следует делиться ни с кем. Скопируйте публичный ключ на удаленный компьютер командой

```
ssh-copy-id -i ~/.ssh/id_edcsa.pub user@hostname
```

Команда запросит у вас пароль на удаленном компьютере и затем скопирует туда публичный ключ. При следующей попытке подключиться к этому компьютеру под данным пользователем вы войдете в систему без запроса пароля.

Для повышения безопасности можно отключить вход по паролям, и тогда в систему смогут войти только пользователи с зарегистрированными ключами. Тогда вам понадобится доступ `root` или `sudo` на целевом компьютере – придется изменить файл `/etc/ssh/sshd_config`. Найдите строку с параметром

```
PasswordAuthentication yes
```

и измените его на `no`. Перезапустите демона SSH, и к компьютеру смогут подключиться только пользователи с ключами. Сделать то же самое для пользователя `root` можно, указав параметры

```
PermitRootLogin no
```

```
PermitRootLogin without-password
```

Вход с паролем для `root` нужно отключать всегда; это самый незащищенный вход, потому что `root` – единственный пользователь, имя которого известно заранее.

Запуск программ

С SSH можно не только открыть сеанс терминала; можно, скажем, запускать программы напрямую. Команда

```
ssh user@computer someprogram --arguments
```

заходит в систему на удаленном компьютере, запускает указанную программу и завершается. Оболочка не открывается, и весь вывод отображается на локальном терминале. Будьте внимательны со специальными символами и кавычками, потому что перед передачей `ssh` команды будут обработаны локальной оболочкой. Так, команда

```
ssh myothercomputer ls *.jpg
```

не сделает того, чего вы ожидали, поскольку локальная оболочка преобразует `*.jpg` в список всех файлов с этим расширением в текущем каталоге и затем передаст `ssh`, которая вернет множество ошибок «Такого файла не существует». Чтобы специальные символы были переданы без интерпретации, нужно их экранировать или брать в кавычки.

```
ssh myothercomputer ls `*.jpg`
```

С помощью SSH можно запускать не только команды в терминале, но и графические программы. Сначала убедитесь, что в `/etc/ssh/sshd_config` есть строка

```
X11Forwarding yes
```

Затем вызовите `ssh` с параметром `-X` или `-Y`. У `-X` есть дополнительные ограничения по безопасности, поэтому пользуйтесь `-Y` только когда `-X` не делает того, что вам нужно, и только в защищенной сети.

```
ssh -X myothercomputer
```

С помощью SSH можно подключиться к компьютеру из любой точки мира, если в маршрутизаторе настроено перенаправление подключений на порт 22 (стандартный порт SSH). Это делает его лакомой целью атак юных хакеров, и ваш системный журнал может быть полон неудачных попыток входа в систему. Хотя аутентификация по ключу предотвратит успех таких атак, ваш компьютер все равно должен с ними бороться. Популярное решение – воспользоваться нестандартным портом. Если ваш маршрутизатор может перенаправлять трафик с одного входящего порта на другой, это просто: выберите любой, но неиспользуемый порт с большим номером, перенаправьте его на порт 22,

➤ Запуск графической программы по SSH; единственное, что выдает удаленный запуск – имя хоста удаленного компьютера в заголовке окна.



ОПАСЕНИЙ

и локальный доступ будет работать так же, как прежде. Просто помните, что вне маршрутизатора нужно указывать другой номер порта, а также внешний IP-адрес или имя хоста компьютера.

```
ssh -p 5678 user@public.ip
```

Если маршрутизатор может перенаправлять только на тот же порт, можно заставить SSH слушать оба порта, и порт 22 будет по-прежнему работать локально – для этого добавьте следующие строки в `sshd_config`:

```
Port 22
```

```
Port 5678
```

Теперь просто запомните, что при подключении извне нужно указывать новый порт. Если вам лень запоминать, можно добавить следующие строки в `~/.ssh/ssh_config`:

```
Host <имя хоста или адрес IP>
```

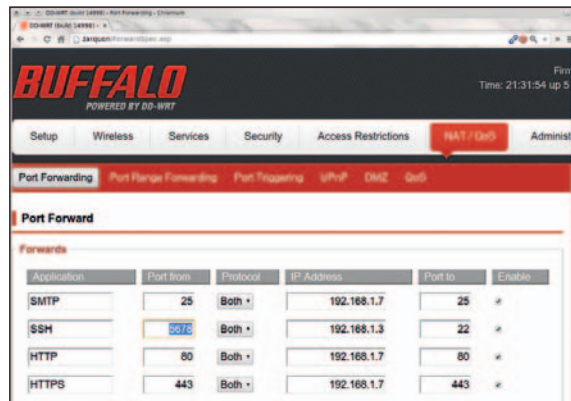
```
Port 5678
```

Любой параметр, указанный после строки `Host`, применяется только к соответствующим хостам. Применяться будут все параметры до следующей строки `Host`, так что перед любыми глобальными настройками ставьте строку `Host *`. Каждый параметр можно задавать только один раз, поэтому настройки для конкретных хостов в этом файле должны идти перед настройками по умолчанию. Эти настройки применяются только к текущему локальному пользователю. Чтобы они применялись ко всем пользователям, воспользуйтесь файлом `/etc/ssh_config`.

Черные списки

Еще один способ борьбы с нежелательными попытками входа по SSH, особенно если вы не можете поменять порт – автоматическая блокировка IP-адресов, попытки входа с которых постоянно завершаются неудачно. Существует несколько программ, которые это делают, одна из самых популярных – `fail2ban` (www.fail2ban.org). Эта программа, как и ее аналоги, сканирует лог-файлы на наличие признаков подозрительной активности. Для SSH это множественные попытки подключения с разными именами пользователя с одного и того же IP-адреса. При обнаружении подозрительного поведения она создает правило брандмауэра, которое блокирует все подключения с этого адреса на заданный период времени. Можно настроить и другие действия, например, отправку письма, но блокирование на уровне брандмауэра используется по умолчанию. Настройки по умолчанию охватывают популярные серверы, в том числе SSH, а также web-сервер `Apache`. Большинству пользователей достаточно установить программу из менеджера пакетов и убедиться, что сервис запускается при загрузке системы. Также можно установить программу вроде `Logwatch` (www.logwatch.org), которая просматривает файлы журналов и ежедневно отправляет суммарный отчет обо всей системной активности.

Программа, на которую стоит обратить внимание всем пользователям SSH – `screen` (www.gnu.org/software/screen). Если вы запускаете программу в удаленной оболочке и сетевое соединение разрывается, оболочка завершится со всеми программами, которые были в ней запущены. `Screen` называют мультиплексором терминалов, т.е. вы запускаете его в оболочке, и на первый взгляд ничего не меняется, но теперь вы находитесь в сессии `screen`. Запустите программу и нажмите `Ctrl+A`, затем `D`, и сессия терминала исчезнет. Выполните команду



```
screen -r
```

и сессия терминала появится снова с запущенной в ней программой, как будто никуда не исчезала. Это особенно удобно с SSH, потому что можно войти в систему, запустить `screen` и выполнять все необходимые действия. Если подключение SSH будет прервано, можно переподключиться и выполнить команду `screen -r`. Это даже не обязательно делать с того же самого компьютера.

Толпа SSH

Обычно с помощью SSH мы подключаемся с одного компьютера к другому, но что если нужно запустить команду на нескольких компьютерах сети? Сделать это можно с `dsh` (www.netfort.gr.jp/~dancer/software/dsh.html.en), разновидностью распределенной оболочки. Основной формат команды таков:

```
dsh -m user1@host1 -m user2@host2 command
```

Хостов можно указать любое количество. С `dsh` вам не придется настраивать аутентификация с помощью ключей, иначе придется набирать огромное количество паролей. Чтобы каждый раз не набирать имена хостов, можно добавить их в группу в файле в каталоге `/etc/dsh/groups/`:

```
user1@host1
```

```
user2@host2
```

```
host3
```

Для последнего хоста работает обычное правило SSH, при котором используется текущее имя пользователя, если оно не указано. Запустите `dsh` следующим образом:

```
dsh -g groupname command
```

Здесь `groupname` – имя файла в `/etc/dsh/groups`; только имя, а не полный путь. Также в файле `/etc/dsh/machines.list` можно задать группу по умолчанию, а затем запустить ту же команду для всех хостов этой группы:

```
dsh -a command
```

Обычно команда запускается на всех компьютерах одновременно. Это самый быстрый способ, но он не слишком хорош, если команды генерируют большой объем выходных данных.

В начале каждой строки указывается имя компьютера, на котором запущена команда, но если выводимых строк много, то здесь легко запутаться. В этом случае укажите параметр `-w`, чтобы `dsh` ожидала завершения команды на каждом компьютере и только потом запускала ее на следующем. Этот и другие параметры можно задать как используемые по умолчанию в `/etc/dsh/dsh.conf`. **LXF**

➤ Если вы хотите открыть доступ по SSH извне, использование нестандартного порта (не 22) позволит удержать хулиганов-скриптописцев на расстоянии.

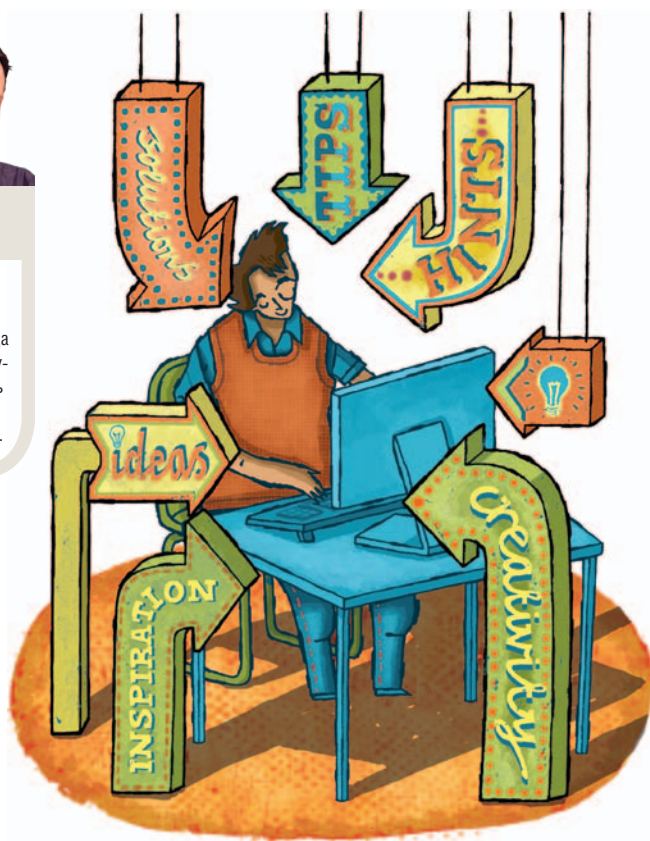
OpenRemote:

Козну Вервлоэсему не нравится, когда в его доме распоряжается «черный ящик», и для своего последнего проекта автоматизации дома он прибегнул к *OpenRemote*.



Наш эксперт

Козну Вервлоэсем пишет об открытии ПО с 2000 года и за это время научился не доверять закрытым системам любых типов.



Большинство решений по автоматизации дома проприетарны. Вам придется заплатить за кучу устройств и программ, тесно переплетенных друг с другом и поддерживающих только заданный набор периферийных устройств. Разумеется, нам в *Linux Format* этот подход не нравится; но в *OpenRemote* мы нашли средство автоматизации, соответствующее нашим потребностям. Контроллер автоматизации с открытым исходным кодом написан на Java, поддерживает несколько протоколов автоматизации, и его можно установить в Linux, Windows или OS X. Кроме того, можно разработать собственные интерфейсы управления, доступные через web-интерфейс и приложение для Android или iPhone. Поэтому с подходящим оборудованием автоматизации потребуется совсем немного времени для того, чтобы вы смогли включать и отключать устройства, пробуждать и выключать компьютеры и выполнять другие действия несколькими щелчками на своем смартфоне.

Установка

Чтобы воспользоваться *OpenRemote*, установите на компьютер программу контроллера. Она написана на Java, так что сперва придется установить Java JRE. Мы установили контроллер на Raspberry Pi с Raspbian Wheezy (см. врезку «*OpenRemote* на Raspberry Pi»), но поддерживаются любые другие дистрибутивы Linux

и даже некоторые устройства NAS на базе Linux, такие как Synology NAS, Netgear ReadyNAS или QNAP NAS. Инструкции по установке на эти устройства можно найти в документации на www.openremote.org. Контроллер *OpenRemote*, разумеется, должен работать постоянно, поэтому для него стоит выбрать экономичное и желательно компактное устройство, например, NAS или Raspberry Pi, которое можно спрятать в подвале.

Загрузите ZIP-файл с *OpenRemote 2.0* и распакуйте его. Затем сделайте скрипт `openremote.sh` исполняемым и запустите его:

```
$ unzip OpenRemote-Controller-2.0.0.zip
$ cd OpenRemote-Controller-2.0.0/bin
$ chmod +x openremote.sh
$ ./openremote.sh run
```

На экране появится множество строк вывода; их придется проанализировать, если появятся ошибки. Если все прошло хорошо, контроллер *OpenRemote* теперь будет слушать команды.

Добавляем интерфейс

Теперь определим поведение и интерфейс для контроллера *OpenRemote*. Для этого нужно создать учетную запись в *OpenRemote Designer* (designer.openremote.org), дизайнере интерфейсов. Начнем с простого примера: создадим кнопку для пробуждения настольного компьютера с помощью Wake-on-LAN [технология, позволяющая удаленно включить компьютер посредством отправки по сети специального пакета, – прим. пер.]. После ввода логина и пароля вы увидите Building Modeler [Конструктор], в котором сможете настроить действия контроллера.

Щелкните на Device > New > New Device [Устройство > Создать > Новое устройство] чтобы добавить новое устройство контроллера, укажите его имя (например, Raspberry Pi, если контроллер работает на нем) и нажмите Next [Далее]. Затем нажмите Add command [Добавить команду], чтобы добавить новую команду, которая будет выполняться контроллером. Задайте ей имя (например, Wake PC – «разбудить компьютер») и выберите протокол; в нашем примере это Wake-On-Lan. Как вы уже видите в выпадающем меню, *OpenRemote* поддерживает приличное количество

«OpenRemote поддерживает приличное количество протоколов.»

протоколов. В нижней части окна введите MAC-адрес настольного компьютера (который можно найти в выводе команды `ifconfig` после `HWaddr`) и широковещательный IP-адрес сети (который можно найти там же после `Bcast`), например, 192.168.0.255. Затем нажмите на Submit [Отправить] и на Finish [Конец]. В левой колонке должно появиться устройство с одной командой.

Итак, команда для пробуждения компьютера определена, но мы пока не определили, как ею управлять. Для этого щелкните по иконке UI Designer [Дизайнер интерфейса] слева сверху (это вторая иконка слева) и создайте новую панель, выбрав New

КОНТРОЛЬ ДОМА

> New Panel [Создать > Новая панель]. Задайте ее имя и тип (например, Android) и нажмите Submit [Отправить]. В окне появится виртуальный экран Android. Перетащите виджет Button [Кнопка] с правой части экрана и зайдите в свойства кнопки справа; нажмите Select [Выбрать] справа от Command [Команда]. Выберите команду Wake PC, которую мы определили ранее, и нажмите OK. Также нужно изменить имя кнопки по умолчанию ("Button") на нечто более информативное. Теперь к кнопке привязана команда, поэтому можно нажать Save [Сохранить] в верхней части интерфейса.

Разбудите свой компьютер

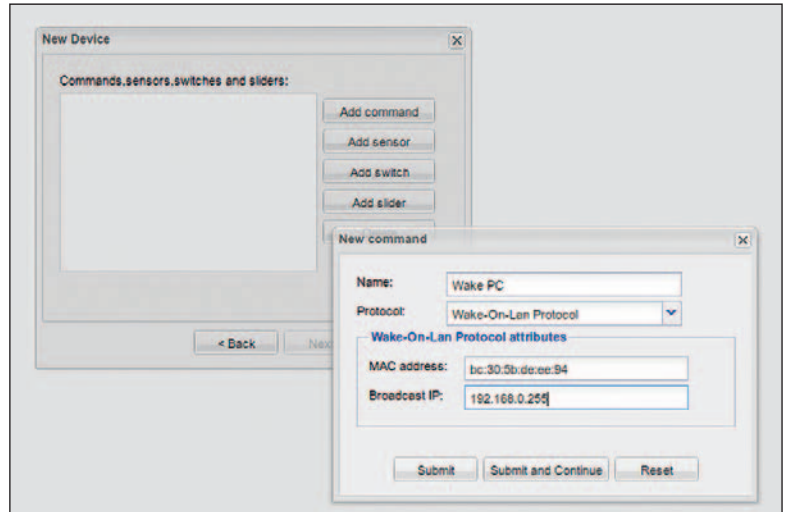
Раз контроллер готов, у нас должна появиться возможность воспользоваться интерфейсом. Просто откройте в браузере адрес <http://IPADDRESS:8080/controller/>, заменив IPADDRESS на IP-адрес компьютера. Введите имя пользователя и пароль для *OpenRemote Designer* и нажмите на кнопку Sync with Online Designer [Синхронизироваться с онлайн-дизайнером]. После появления сообщения "Sync Complete [Синхронизация завершена]" откройте адрес <http://IPADDRESS:8080/webconsole> и нажмите на кнопку Search [Поиск], после чего консоль должна автоматически найти контроллер.

Нажмите на стрелку справа от адреса контроллера, введите имя своей панели в поле Default Panel Name [Имя панели по умолчанию] и нажмите Save [Сохранить]. Если теперь щелкнуть по адресу контроллера в списке, должна открыться панель с кнопкой, которую мы добавили на виртуальный экран смартфона. Нажатие на нее заставит ваш компьютер проснуться.

Если это не работает, возможно, проблема в wake-on-LAN, а не в *OpenRemote*. Прежде всего проверьте, правильно ли ввели MAC-адрес и широковещательный IP-адрес. Также, возможно, придется включить **Wake-on-LAN** в BIOS, или даже обновить BIOS. Для проверки функции воспользуйтесь командой `wakeonlan`, запустив ее с вашим MAC-адресом, чтобы проверить это вне *OpenRemote*.

Приложение для Android

При помощи web-консоли можно управлять контроллером *OpenRemote* с любого устройства, на котором есть браузер; но кроме этого, есть и отдельные приложения для Android и iOS. Для Android установите приложение *OpenRemote* в Google Play. При запуске приложения оно автоматически обнаружит контроллер, если телефон с Android подключен к той же сети, что и контроллер, при условии, что не была отключена возможность автоматического обнаружения [Auto Discovery]. Внизу выберите панель, которая вам нужна. В этой статье мы определили всего одну панель, но их может быть и несколько для разных интерфейсов одного и того же контроллера – например, один для телефона с Android и один для iPad. Наконец, нажмите Done [Готово], после чего в приложении откроется панель, созданная в *OpenRemote Designer*. При следующем запуске приложения она откроется сразу. Теперь вы сможете разбудить свой компьютер со смартфона через программу *OpenRemote*.



Будить-то компьютер с телефона мы научились, но как его выключать? Это тоже можно сделать в *OpenRemote*: нужно задать команду, которая выполняет скрипт оболочки.

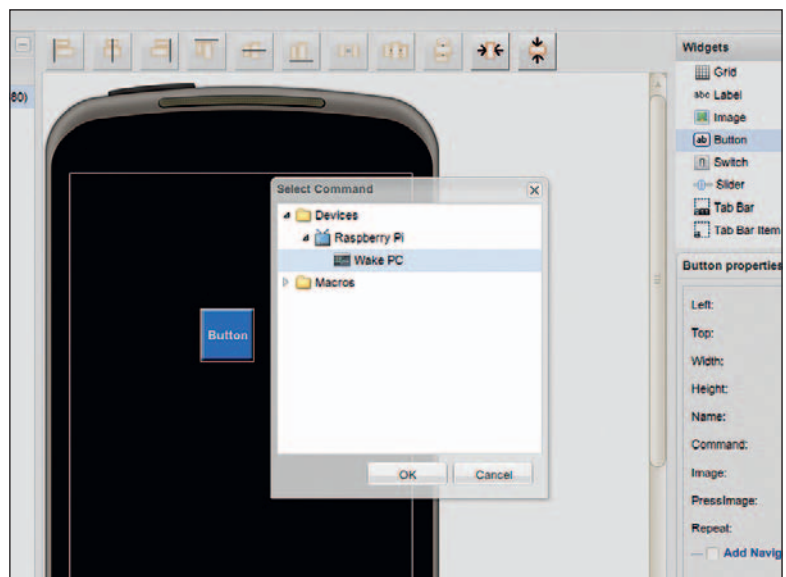
Создадим на контроллере скрипт `shutdown_computer.sh`, который выключает компьютер. В качестве аргумента ему нужно передать IP-адрес компьютера:

```
#!/bin/bash
ssh -t openremote@$1 sudo shutdown -h now
```

Параметр `-t` необходим потому, что `sudo` требуется терминал. Затем сделаем скрипт исполняемым:

```
$ sudo chmod +x /home/pi/shutdown_computer.sh
```

» Хотите разбудить компьютер? Для этого есть команда!



» Если вам надоел пульт управления телевизором, создайте свой собственный интерфейс удаленного управления в дизайнера интерфейсов.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Скорая помощь

Для проверки интерфейса, созданного в дизайнера интерфейсов, не обязательно иметь установленный контроллер *OpenRemote*. По щелчку на самой правой иконке вверху слева в *OpenRemote Designer* откроется URL, который можно использовать как адрес контроллера в своем приложении для проверки.

Скорая помощь

Если синхронизация настроек контроллера *OpenRemote* с учетной записью *OpenRemote Designer* не работает, всегда можно экспортировать конфигурацию в дизайнер в zip-файл и загрузить его в контроллер *OpenRemote*. Выберите Offline [Без синхронизации] на домашней странице контроллера.

➤ Приложение *OpenRemote* автоматически обновит контроллер в вашем подвале, поэтому не нужно узнавать его IP-адрес.

Теперь на компьютере, который нужно выключать удаленно, выполните команду `sudo visudo` и добавьте следующие строки:

```
Cmnd_Alias POWER = /sbin/reboot,/sbin/shutdown
openremote ALL=(ALL) NOPASSWD: POWER
```

Это гарантирует разрешение пользователю `openremote` перезагружать и выключать компьютер без ввода пароля. Конечно, этого пользователя еще нужно создать, таким образом:

```
$ sudo adduser openremote
```

Теперь, так как мы хотим входить в систему от имени пользователя `openremote` без ввода пароля (так как наш скрипт нужно запускать вручную), придется сгенерировать пару ключей без ключевой фразы на контроллере и скопировать публичный ключ SSH пользователя с контроллера на наш компьютер. Для этого скомандуйте

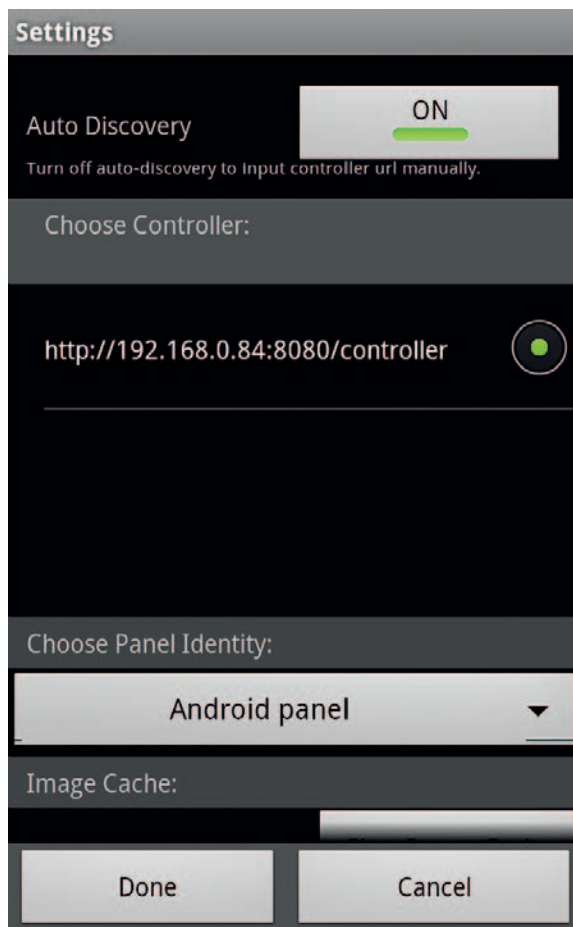
```
$ ssh-keygen
$ ssh-copy-id -i ~/.ssh/id_rsa.pub openremote@PC
```

Для начала попробуем запустить скрипт с нужным IP-адресом:

```
$ ./shutdown_computer.sh IPADDRESS
```

Если это работает, создайте в *Open Remote Designer* новую команду, выполняющую этот скрипт. Задайте ее имя, например, "Shutdown PC", выберите в качестве протокола Shell execution protocol и введите полный путь до скрипта, а также IP-адрес как аргумент. На данный момент этот протокол поддерживает только один аргумент скрипта. Нажмите Submit.

Пока мы удаленно управляли только компьютером, но, надеюсь, у вас дома есть не одни компьютеры... при желании управлять другими устройствами можно воспользоваться протоколом автоматизации дома, например, X10, Z-Wave или KNX; все они прямо или косвенно поддерживаются *OpenRemote*.



Мы покажем, как опосредованно управлять переключателями в X10 из *OpenRemote*. Таким образом можно автоматически включать и отключать кофеварку или свет в доме. Единственное, что для этого нужно – контроллер X10 и модули X10. Контроллер X10 отправляет сигналы по линиям питания, а модуль X10 получает эти сигналы и переключает устройство, подсоединенное к модулю. Подключив контроллер X10 к нашему компьютеру с контроллером *OpenRemote*, мы можем управлять нашими устройствами X10 централизованно из того же интерфейса *OpenRemote*, где находятся другие команды.

Компиляция и использование Mochad

У нас есть контроллер X10 Marmitek XM15Pro, который поддерживается *Mochad* (Multiple Online Controllers for Home Automation Daemon – демон нескольких онлайн-контроллеров для автоматизации дома). Его нет в репозиториях Debian, поэтому вам придется загрузить и скомпилировать его самим. К счастью, сделать это нетрудно:

```
$ sudo apt-get install libusb-1.0-0-dev
$ tar xvzf mochad-*.tar.gz
$ cd mochad-*
$ ./configure
$ make
$ sudo make install
```

Пользоваться *Mochad* довольно легко. При подключении контроллера X10 к компьютеру через USB *Mochad* автоматически запускается как демон. В файле `/var/log/messages` должны появиться следующие записи:

```
Oct 21 15:11:45 raspberrypi mochad[4962]: starting
Oct 21 15:11:45 raspberrypi mochad[4963]: Found CM15A
Oct 21 15:11:45 raspberrypi mochad[4963]: In endpoint 0x81,
Out endpoint 0x02
```

После запуска *Mochad* можно управлять демоном командой `netcat` через порт 1099:

```
$ nc localhost 1099
```

Теперь можно вводить команды X10, например, `st` для просмотра состояния и `pl b2 on` для включения модуля с кодом X10 `B2 on`. У каждого модуля X10 должен быть уникальный код дома [House Code] от A до P и код модуля [Unit Code] от 1 до 16. *Mochad* также распознает некоторые общие команды, например, `all_lights_off` [выключить свет везде].

Если это заработало, привяжите команды *Mochad* к *OpenRemote*. Откройте дизайнер *OpenRemote* и добавьте новую команду, например, `Coffee maker on` [Включить кофеварку]. Задайте TCP/IP (не X10!) в качестве протокола, localhost в качестве IP-адреса и 1099 в качестве порта. Введите нужную команду *Mochad*, например, `pl b2 on`, в поле Command. Нажмите Submit. Теперь добавьте виджет Button в дизайнера интерфейса и привяжите созданную команду к виджету. Нажмите Save и скопируйте конфигурацию нового контроллера в контроллер *OpenRemote*. Если все получится, вы сможете включить кофеварку со смартфона с Android.

Информацию о поддержке *OpenRemote* других устройств X10 можно найти в документации. Некоторые устройства поддерживаются по умолчанию; в этом случае в качестве протокола для команд нужно выбирать X10. Однако протокол TCP/IP является более гибким, поскольку его можно использовать также и с любыми устройствами или программами, которыми можно управлять в *netcat*.

Последний метод, который мы продемонстрируем – получение данных с сайта командой HTTP и создание отображающего их датчика на панели *OpenRemote*. Для примера воспользуемся

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

сервисом Weather2 (www.myweather2.com). Создайте бесплатную учетную запись на этом сайте и обратите внимание на уникальный код доступа в разделе Developer Zone [Разработчикам]. Теперь, если вы хотите узнать погоду в Великобритании в местности с почтовым индексом BA1 (это штаб-квартира LXF), откройте следующий адрес: www.myweather2.com/developer/forecast.ashx?uac=UNIQUEACCESSCODE&output=xml&query=BA1. Он вернет XML-файл с прогнозом погоды на два дня.

Если вы хотите воспользоваться этим прогнозом погоды в *OpenRemote*, создайте новую команду, задайте ее имя (например, «Current temperature [Текущая температура]») и выберите HTTP в качестве протокола. В поле URL введите URL прогноза.

Так как web-сервер возвращает XML-файл, здесь можно ввести выражение XPath для добычи нужной информации. Например, выражение для извлечения текущей температуры будет таким: `//current_weather/temp`. Затем выберите интервал опроса, например, «30m» для опроса раз в полчаса, и нажмите Submit.

Создаем датчик

Мы обзавелись командой для извлечения текущей температуры, но ее нужно показать в интерфейсе пользователя. Для этого создайте новый датчик, связываемый с командой. Нажмите New sensor [Новый датчик], задайте ему имя Current temperature и выберите команду, которую нужно с ним связать. В качестве типа команды выберите Custom [Пользовательская]. Нажмите Submit, и после этого датчик появится в левой колонке.

Теперь перейдите в дизайнер интерфейсов и добавьте метку со статичным текстом (например, **Температура в г. Бат (°C)**) и еще одну метку, которую вы свяжете с датчиком Current temperature. Для этого щелкните на Sensor [Датчик] и выберите подходящий датчик. Наконец, сохраните конфигурацию нового контроллера и синхронизируйте ее с контроллером *OpenRemote*. Теперь на смартфоне появится текущая температура.

OpenRemote на Raspberry Pi

Raspberry Pi – идеальный соратник *OpenRemote*: он невелик в размерах и не тратит много энергии, поэтому прекрасно может работать постоянно где-нибудь в подвале и автоматизировать ваш дом. Для этой статьи мы установили на SD-карту образ Raspbian Wheezy. После этого нужно установить необходимую версию Java:

```
$ sudo apt-get install icedtea-6-jre-cacao
```

Может потребоваться время, так как устанавливается множество зависимостей. По окончании установки Java просто установите *OpenRemote* согласно нашему описанию в основной статье. Добавим только одно: если *OpenRemote* должен

запускаться автоматически при загрузке Raspberry Pi, измените файл `/etc/rc.local`.

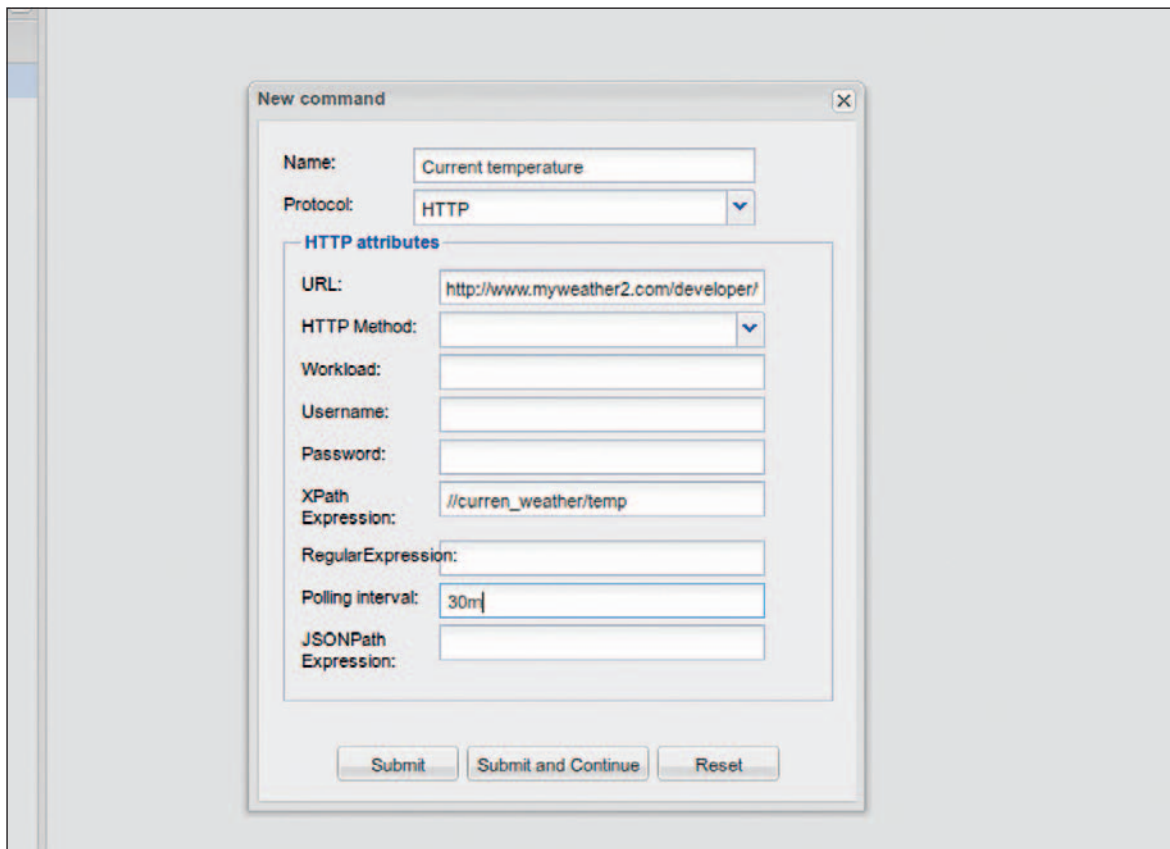
Это не самое общее решение, но оно проще, чем создавать целый скрипт `init`. Просто добавьте две следующие строки перед `exit 0`:

```
cd /home/pi/OpenRemote-Controller-2.0.0/bin
./openremote.sh start
```

Учтите, что для крошки Raspberry Pi *OpenRemote* – ресурсоемкая программа, и будет запускаться медленно (на весь процесс запуска уходит 50 секунд), но после этого программа полностью готова к использованию.

OpenRemote легко интегрируется со многими другими устройствами и программами – загляните в документацию, чтобы узнать об этом подробнее. Например, с помощью панели *OpenRemote* можно управлять медиа-проигрывателем XBMC на некоторых телевизорах Samsung; *OpenRemote* можно использовать с системами сигнализации избирательного цифрового вызова, инфракрасными приемниками... список можно продолжить.

OpenRemote все еще находится в разработке и не является готовым и быстрым решением, однако есть прекрасное сообщество экспериментаторов, которые добавляют поддержку новых устройств, и когда вы будете читать эту статью, у вас под рукой появится еще больше новых возможностей превратить свое жилище в «умный дом». LXF



» Благодаря *OpenRemote* каждая веб-страница становится датчиком.

Скорая помощь

OpenRemote также знакома концепция выключателя. Для него необходимы команды включения и выключения (например, наши команды для пробуждения и выключения компьютера), а также датчик, с которого можно прочесть текущее состояние. После их создания выберите New > New Switch [Создать > Новый выключатель] в конструкторе [Building Modeler] и свяжите три этих компонента с выключателем.



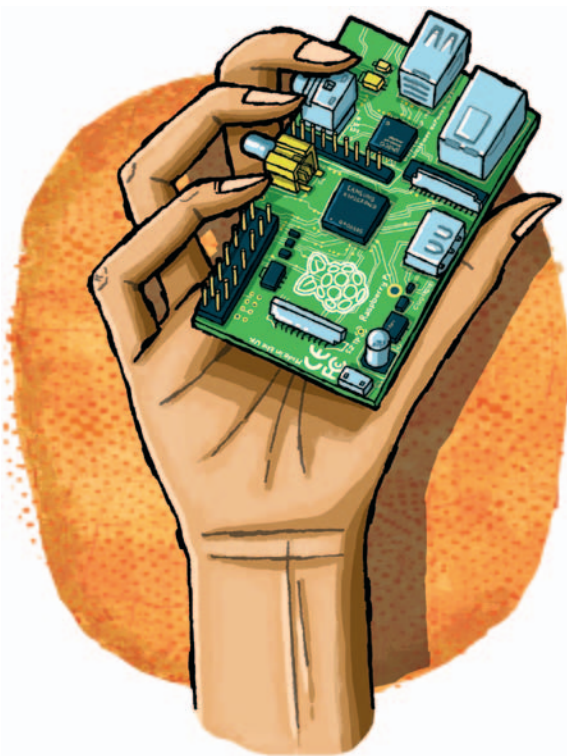
Raspberry Pi: СВОЙ

Теперь, чтобы взяться за дело, не нужно ждать, пока Raspberry Pi загрузится. **Джон Лэйн** создает пользовательский дистрибутив со всем необходимым.



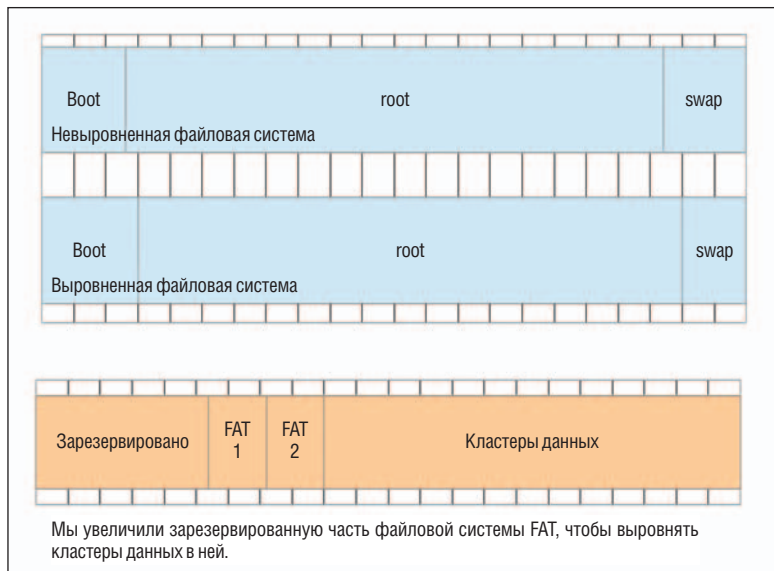
Наш эксперт

Забредя в Linux еще в 1994 году, **Джон Лэйн** так в нем и остался, и забыл, когда в последний раз пользовался Windows. Он пишет и консультирует по Linux и открытому ПО.



На этом уроке мы попробуем создать собственный дистрибутив для Raspberry Pi, где будет только то, что вам нужно. Вы можете захотеть включить в него другие пакеты, менять настройки, возможно, даже изменить ядро. Образ дистрибутива можно будет записать на SD-карту и загрузить с нее Pi или загрузить его на компьютере через эмулятор. Создавать свой

► **Выравнивание по блокам стирания дает нам претолченную SD-карту.**



образ можно прямо на Pi, но это будет очень медленно, поэтому мы заодно рассмотрим, как скомпилировать и даже запустить ARM-код на вашем компьютере.

Конечно, настройки можно было бы делать прямо в работающей системе, но это опасно, потому что из-за одной ошибки все может рухнуть. Вдобавок эти шаги довольно сложно воспроизвести и трудно автоматизировать.

Первым делом нужно обзавестись подходящими инструментами. Для нашего урока мы возьмем Arch Linux, так как это прекрасный дистрибутив, доступный и для ПК, и для Pi. Будем предполагать, что используется последняя официальная версия Arch Linux и что она обновлена. Загрузите Arch Linux на своем Pi и убедитесь, что необходимые утилиты и другие пакеты установлены:

```
# pacman -Syu
# pacman -S base-devel python2 git parted dosfstools
```

Выбираем себе приключение

Первый шаг на пути к собственному дистрибутиву – решение, что в нем должно присутствовать. Разумный вариант для начала – базовая группа пакетов Arch Linux (base). Берите ее и добавляйте и удаляйте необходимые пакеты. Ненужные пакеты ядра стоит удалить и добавить соответствующие пакеты Raspberry Pi.

Сборка нового ядра необходима, если требуется добавить новые возможности. Также стоит отключить ненужные опции ядра, чтобы уменьшить его размер (учитывая ограниченные ресурсы памяти у малыша Pi). Учтывая то, что компиляция ядра обычно занимает более десяти часов, нам нужен альтернативный метод. Одна из альтернатив – использование кросс-компилятора; можно также использовать *distcc*, но с ним все равно понадобится кросс-компилятор, потому что «сервер» *distcc* должен генерировать код ARM. Я решил выполнить кросс-компиляцию ядра, и на нашем уроке опишу оба метода.

В идущих ниже командах для получения списка пакетов группы base используется *Pacman*, менеджер пакетов Arch Linux. Затем удаляются ненужные пакеты и добавляются ядро RPi и прошивка. Для добавления или удаления всего прочего в соответствии со своими потребностями (например, вы можете захотеть добавить *openssh*), употребите свой любимый текстовый редактор.

```
$ pacman -Sg base | awk '{print $2}' | grep -v "^(linux|kernel)" |
tr '\n' ' ' > base_packages
$ sed -i -e 's/\/linux-raspberrypi linux-headers-raspberrypi
raspberrypi-firmware/' base_packages
```

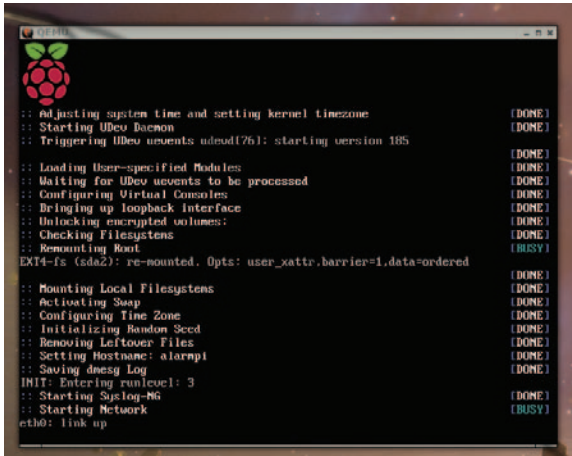
Вооружившись списком пакетов, установим пакеты для нового образа. С утилитой *mkarchroot* сделать это легко. Она установит пакеты в подкаталог, и его вы затем преобразуете в образ диска.

```
# mkarchroot archroot $(cat base_packages)
```

Создается новый подкаталог **archroot**, содержащий все необходимое для полной системы. Затем его можно сделать корневым, и это должно работать; на практике, это придется делать при любых изменениях, например, для задания пароля *root* или для добавления учетных записей пользователей.

Мы создадим образ системы в файле. Его можно создать и прямо на SD-карте, но иметь образ полезно, потому что его

ДИСТРИБУТИВ



► Почти как настоящее: QEMU эмулирует Pi из нашего образа.

можно быстро перезаписать на карту или загрузить в эмуляторе даже без записи на карту. У ядра есть возможность «петли [loop]», при которой файл рассматривается как физический диск, который можно разбить на файловые системы. Для такого доступа к файлу в каталоге `/dev` создается обычный файл устройства.

Образ должен быть достаточно большим для размещения файловых систем, но и достаточно маленьким, чтобы поместиться на SD-карту. Мы создадим образ размером 2 Гиб [здесь и далее используются двоичные приставки; их отличие от соответствующих десятичных в том, что умножение производится на степени двойки; таким образом, 1 КиБ (кибибайт) = 2¹⁰ = 1024 байта, 1 МиБ (мебибайт) = 2²⁰ = 1 048 576 байт, 1 ГиБ (гибибайт) = 2³⁰ = 1 073 741 824 байта, – прим. пер.], такого же размера, как и официальный образ. Убедитесь, что модуль ядра `loop` загружен, создайте файл и свяжите с ним устройство «петли»:

```
# modprobe loop
$ truncate -s 2G myimage
$ device=$(losetup -f)
# losetup $device myimage
```

Мы пользуемся `losetup` дважды: для выделения имени устройству «петли», обычно `/dev/loop0`, и для его создания (имя устройства хранится в переменной, чтобы обратиться к нему позже). Разделы на образе должны быть разбиты особым образом: первый раздел с файловой системой FAT16 должен содержать загрузочные файлы и образ ядра. Второй раздел используется для корневой файловой системы и должен быть типа `ext4`. Если необходим раздел подкачки, его можно создать третьим. Загрузочная прошивка Raspberry Pi ищет таблицу разделов главной загрузочной записи типа MS-DOS (в отличие от BIOS ПК, она не запускает загрузчик из главной загрузочной записи). Для создания таблицы разделов на файле образа используйте команду `parted`:

```
# parted -s $device mktable msdos
```

При создании разделов стоит выравнивать их по блокам стирания на SD-карте. Делать это не обязательно, но рекомендуется.

Работа с chroot в вашем образе

Иногда полезно поработать внутри образа файловой системы, например, **archroot**, которую мы создали в этом разделе. Для этого можно воспользоваться **chroot**, но сначала стоит смонтировать некоторые части файловой системы. Вот маленький скрипт, чтобы войти в `chroot`:

```
#!/bin/bash
mkdir -p $1/(dev/pts,proc)
mount proc -t proc $1/proc
mount devpts -t devpts $1/dev/pts
chroot $1 /usr/bin/env -i
TERM="$TERM" /bin/bash --login
umount $1/(dev/pts, proc)
```

Рекомендуемый размер области стирания для большинства карт – 4 МиБ, но можно проверить его и для конкретной карты (в байтах):

```
$ cat /sys/class/block/mmcblk0/device/preferred_erase_size
```

Если этот размер равен 4 МиБ, то выравнивание производится через каждые 8192 сектора (в секторе 512 байт, поэтому число секторов – 4 МиБ/512). Значит, разделы должны начинаться с сектора, кратного 8192: в первом блоке стирания 8192 сектора (0..8191), во втором – следующие 8192 сектора (8192..16383) и т.д. Сектор 0 резервирован для таблицы разделов, поэтому первый раздел должен начинаться со следующего выровненного сектора, это сектор 8192. Хороший размер для загрузочного раздела – 40 МиБ, то есть 81 920 секторов. Этого достаточно для размещения загрузочных файлов, и к тому же раздел заканчивается на границе блока стирания. Создайте загрузочный раздел, начиная с сектора 8192 и заканчивая сектором 90111 (8192+81920-1):

```
# parted -s $device unit s mkpart primary fat32 8192 90111
```

Если вам нужен раздел подкачки, выделите под него немного места (о необходимости раздела подкачки на SD-карте стоит подумать из-за медленной скорости и ограниченного количества операций записи носителя). Например, мы создадим раздел подкачки размером 256 МиБ (это 524 288 сектора или 64 блока записи).

Наш образ размером 2 Гиб содержит 4 194 304 сектора (2 Гиб / 512 = 4 194 304). Так как нумерация секторов начинается с 0, то последним сектором будет 4 194 303. Для корневого раздела можно использовать все пространство между разделами `boot` и `swap`, он начнется с выровненного сектора 90 112 и закончится на границе выравнивания, оставив разделу подкачки 56 МиБ. Создадим их:

```
# parted $device unit s mkpart primary ext2 90112 3670015
# parted $device unit s mkpart primary linux-swaps 3670016 4194303
```

При желании можно вывести таблицу разделов командой `parted -s $device unit s print`. Теперь создадим файловые системы. Устройство «петли» нужно пересоздать заново, чтобы для разделов были созданы файл устройств (это делает параметр `-P`):

```
# losetup -d $device
# device=$(losetup -f)
# losetup -P $device myimage
```

Подумаем о выравнивании внутренней структуры файловых систем. В случае файловой системы FAT нужно знать размер ее таблицы размещения файлов (File Allocation Table – FAT), для чего придется создать файловую систему:

Скорая помощь

Если у вас 32-битная система, вы можете воспользоваться встроенным кросс-компилятором `arm-bcm2708linux-gnueabi`, который входит в состав утилит Raspberry Pi. См. [git://github.com/raspberrypi/tools](https://github.com/raspberrypi/tools).

Скорая помощь

Используйте все возможности многоядерного процессора: команда `make` с параметром `-j` заставляет его запустить несколько одновременно выполняющихся заданий. Для Core i7 с 4 ядрами (8 нитями) используйте значение 8.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Скорая помощь



Если у вас нет *Yaourt*, его исходный код можно найти в пользовательском репозитории (AUR – Arch User Repository). Можно не собирать его самому и воспользоваться пакетом из неофициального [archlinuxfr] репозитория. См. <http://archlinuxfr/yaourt-en>.

Скорая помощь



Дополнительные утилиты для Raspberry Pi имеются на <https://github.com/johnlane/rpi-utils>.

➤ Раздвоение личности: x86 превращается в ARM!

```
mkfs.vfat -l -F 16 -n boot -s 16 -v ${device}p1 | grep "FAT size"
```

Структура файловой системы FAT показана на схеме. Для выравнивания измените размер зарезервированного пространства с целью выровнять начало области данных. Размер этого пространства должен соответствовать рекомендуемому размеру области стирания за вычетом размеров двух таблиц выделения файлов. Если размер FAT равен 32 секторам, то размер зарезервированного пространства будет равен $8192 - (2 * 32) = 8128$ секторам. Снова создайте файловую систему с использованием этой информации:

```
mkfs.vfat -l -F 16 -n boot -s 16 -R 8128 -v ${device}p1
```

Для корневого раздела используйте файловую систему ext4. Благодаря размеру блока в 4 КиБ она выравнивается по блокам стирания. Создавайте ее без журналирования, так как это уменьшит количество операций записи на SD-карту:

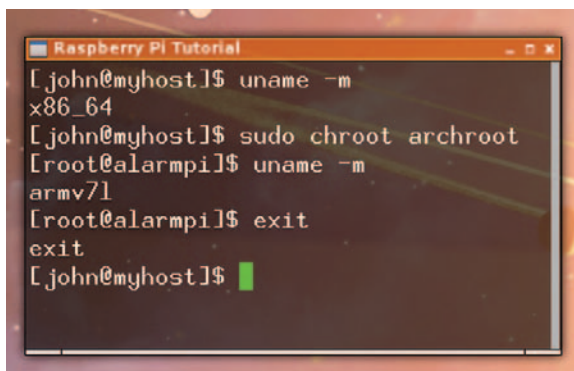
```
$ mkfs.ext4 -O has_journal -L root ${device}p2
```

Смонтируйте файловые системы и скопируйте файлы в соответствующее место из каталога **archroot**. Загрузочный раздел монтируется в каталог **/boot** на корневом разделе, чтобы загрузочные файлы поместились на верный раздел:

```
# mount ${device}p2 /mnt
# mkdir /mnt/boot
# mount ${device}p1 /mnt/boot
# cd archroot ; cp -a * /mnt
# umount /mnt/boot /mnt
```

Теперь запишите образ на SD-карту (если на ней находится корневая система Pi, нужно скопировать изображение на другой компьютер через кард-ридер):

```
dd if=myimage of=/dev/mmcblk0 bs=4M
```



Выключите Pi, вставьте новую карту и загрузитесь. Либо скопируйте образ на компьютер и запустите его в эмуляторе...

Пакет *linux-raspberry*, который мы включили в образ, содержит официальный образ ядра, но изменить его довольно просто. Для его превращения в исполняемый образ ядра нам понадобятся исходный код ядра и компилятор.

Обычно компилятор формирует исполняемый код для того процессора, на котором запущен; но с помощью кросс-компилятора можно создавать исполняемые файлы для различных процессоров. То есть можно воспользоваться ресурсами более мощного «железа» x86 и скомпилировать код для архитектуры ARM Raspberry Pi гораздо быстрее, чем на самом Pi (компиляция ядра 3.2.27 на Pi занимает несколько часов, а с кросс-компилятором на Intel i7 на нее уходит две с половиной минуты!). Поэтому на время отложите Pi в сторону и приготовьтесь к компиляции на своем компьютере. «Родной» компилятор для нашей архитектуры – *gcc*, а версия кросс-компилятора для ARM называется *arm-linux-gnueabi-gcc*. Его нет в репозиториях, но его просто собрать с *Yaourt* (еще одна утилита для работы с репозиториями):

```
yaourt -S arm-linux-gnueabi-gcc
```

Не отходите от терминала, чтобы ответить на многочисленные вопросы *Yaourt* (выберите «нет» на вопрос об изменении PKGBUILD и «да» на вопрос о том, хотите ли вы собрать или установить пакет; сборка проходит по этапам, поэтому всегда выбирайте «да» для замены конфликтующих пакетов).

Привет, мир!

Перед тем как двигаться дальше, пожалуй, стоит проверить кросс-компилятор. По моде истинных программистов мы напишем короткий пример «Привет, мир». Создайте новый файл **hello.c**:

```
#include <stdio.h>
int main ()
{
    printf("Hello, World!\n");
    return 0;
}
```

Скомпилируйте его для ARM и проверьте, что тип указан верно:

```
$ arm-linux-gnueabi-gcc -o hello hello.c
```

```
$ file hello
```

```
hello: ELF 32-bit LSB executable, ARM, version 1 (SYSV),
dynamically linked (uses shared libs), for GNU/Linux 2.6.27, not
stripped
```

Если скопировать файл «hello» на Pi, он должен там запуститься:

```
$/hello
```

```
Hello, World!
```

Если все прошло успешно, то у нас есть кросс-компилятор, и мы можем перейти к компиляции пользовательского ядра.

```
$ git clone --depth 1 git://github.com/raspberrypi/linux.git
```

```
$ cd linux
```

```
$ ssh root@alarmpi zcat /proc/config.gz > .config
```

```
$ make -j 8 ARCH=arm CROSS_COMPILE=arm-linuxgnueabi-
menuconfig -k
```

Мы загружаем исходники ядра из github. Благодаря параметру **--depth 1** загружается только последняя версия, а не вся история изменений – так гораздо быстрее. Затем мы копируем настройки текущего ядра Raspberry Pi в файл **.config** и открываем его в редакторе меню ядра. В нем можно изменить конфигурацию ядра согласно своим пожеланиям. Закончив, выйдите из редактора, сохранив обновленный файл **.config**. Теперь все готово к сборке ядра и модулей (которые мы установим в подкаталог):

```
$ make -j 8 ARCH=arm CROSS_COMPILE=arm-linuxgnueabi- -k
```

```
$ mkdir -p modules
```

Последовательность загрузки Pi

Raspberry Pi загружается не так, как обычный компьютер. Загрузку на Pi начинает графический процессор (а не процессор ARM, как можно было бы предположить). После включения питания графический процессор запускает первый загрузчик (прошивку ПЗУ). Он загружает загрузчики следующих этапов из первого раздела SD-карты (отформатированном в FAT16). В кэше L2 графического процессора загружается и запускается загрузчик второго этапа **bootcode.bin**.

Он, в свою очередь, загружает в оперативную память загрузчик третьего этапа

loader.bin, который считывает прошивку графического процессора **start.elf**. Настраивается этот процесс в *дополнительных* файлах **config.txt** и **cmdline.txt**. Наконец, **start.elf** загружает образ ядра Linux **kernel.img** в пространство памяти процессора ARM, и тот активируется. При этом начинается выполнение образа, а за ним следует стандартный процесс загрузки Linux.

Традиционная главная загрузочная запись (MBR, сектор 0) SD-карты содержит только информацию о таблице разделов. В отличие от ПК, код главной загрузочной записи не выполняется.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.



Erlang: И снова

Андрей Ушаков считает, что одного практикума по многозадачности маловато — и проводит второй.



Наш эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

В прошлый раз в нашем практикуме по многозадачности мы решили создать многозадачные версии таких популярных в программировании функций, как **map** (операция отображения) и **reduce** (операция свертки). Эти функции были выбраны не случайно: их обычные, не многозадачные реализации настолько просты, что при создании их многозадачных версий не зачем сильно отвлекаться на детали, не связанные с многозадачностью. Простоту их мы показали в прошлый раз, создав их обычную (не многозадачную) реализацию. Также мы создали простую многозадачную версию функции **map**, использующую для отображения каждого элемента свой рабочий процесс. При таком подходе (по процессу на элемент) создание многозадачной версии функции **reduce** не имеет смысла. Для создания многозадачной версии функции **reduce** (и более гибкой многозадачной версии функции **map**) надо разбивать данные на порции и эти порции обрабатывать параллельно, после чего формировать итоговый результат. На этом мы и остановились в прошлый раз.

Снова рассмотрим вспомогательные функции, созданные в прошлый раз для работы с порциями данных. Эти функции импортируются из модуля **parallel_common**. Функция **calc_portion_count/2** позволяет вычислить количество порций данных по заданным размерам порции и исходного списка с данными:

```
calc_portion_count(TotalSize, PortionSize)
when TotalSize rem PortionSize == 0 -> TotalSize div PortionSize;
calc_portion_count(TotalSize, PortionSize)
when TotalSize rem PortionSize /= 0 -> (TotalSize div PortionSize) + 1.
```

Пара функций **prepare_data/2** (интерфейсная функция) и **prepare_data/3** (функция, реализующая данную функциональность) разбивают исходные данные на порции:

```
prepare_data(_PortionSize, []) -> [];
prepare_data(PortionSize, SourceList) -> prepare_data(0,
PortionSize, SourceList, []).
prepare_data(Index, PortionSize, SourceList, PreparedData)
when length(SourceList) =< PortionSize ->
lists:reverse([Index, SourceList] ++ PreparedData);
prepare_data(Index, PortionSize, SourceList, PreparedData) ->
{Portion, Rest} = lists:split(PortionSize, SourceList),
prepare_data(Index + 1, PortionSize, Rest, [Index, Portion] ++
PreparedData).
```

Исходный список разбивается на список пар (кортежей), состоящих из индекса порции и собственно самой порции с данными.

Теперь мы готовы двигаться дальше. В многозадачной версии функции **reduce** у нас будет один главный процесс и несколько вспомогательных рабочих процессов. Главный процесс (в котором мы иницируем выполнение нашей версии функции **reduce**) отвечает за разбиение данных на порции, создание необходимого количества рабочих процессов и заданий для них, сбор результатов работы от всех рабочих процессов и, наконец, свертку результатов работы вспомогательных процессов в итоговый результат. Как и при реализации функции **parallel_map: simple_pmap/2**, мы используем «одноразовые» рабочие процессы. При таком подходе можно раздать задания сразу же при создании процессов. А рабочая функция «одноразовых» процессов крайне проста:

вычислить результат свертки для порции и отослать его главному процессу. После создания всех «одноразовых» рабочих процессов все взаимодействие с ними сводится только к получению от них результатов и сохранению этих результатов в промежуточном буфере (в массиве). Для этого мы можем использовать функцию **parallel_common: collect_result/2**, только вместо сохранения отдельных элементов после отображения (что мы делали, когда реализовывали простейший многозадачный вариант функции **map**) мы будем сохранять результаты свертки для группы элементов. Подведя итог вышесказанному, мы можем написать рабочую функцию главного процесса (она же точка входа в многозадачную версию функции **reduce**) следующим образом:

```
portion_reduce(_Fun, [], {InitValue, _PortionInitValue}, _
PortionSize) -> InitValue;
portion_reduce(Fun, SourceList, {InitValue, _PortionInitValue},
PortionSize)
when length(SourceList) =< PortionSize ->
lists:foldl(Fun, InitValue, SourceList);
portion_reduce(Fun, SourceList, {InitValue, PortionInitValue},
PortionSize) ->
process_flag(trap_exit, true),
MasterPid = self(),
PortionCount = parallel_common:calc_portion_
count(length(SourceList), PortionSize),
PreparedData = parallel_common:prepare_data(PortionSize,
SourceList),
lists:foreach(fun({Index, Portion}) -> spawn_link(fun() ->
portion_worker(Fun, Portion, PortionInitValue, Index,
MasterPid) end) end, PreparedData),
EmptyStorage = array:new({Size, PortionCount}, {fixed, true},
{default, none}),
FullStorage = parallel_common:collect_result(EmptyStorage,
PortionCount),
process_flag(trap_exit, false),
lists:foldl(Fun, InitValue, array:to_list(FullStorage)).
```

Эта функция содержит три варианта. Первый обрабатывает наиболее тривиальный случай, когда исходный список пуст, и мы возвращаем начальное значение операции свертки. Второй вариант обрабатывает ситуацию, когда количество элементов в списке не больше размера порции; тогда нет смысла распараллеливать выполнение функции, и операция свертки выполняется при помощи функции **lists:foldl/3**. И, наконец, последний вариант обрабатывает наиболее общий случай, который и является многозадачным. При этом не стоит забывать, что при помощи рабочих процессов мы формируем только результаты операции свертки по подгруппам; финальную операцию свертки промежуточных результатов мы выполняем в конце рабочей функции главного процесса. И, конечно, мы помним, что функция **portion_reduce/3** является экспортируемой функцией модуля **parallel_reduce**.

Как говорилось в предыдущей статье, при реализации параллельного варианта функции **reduce** нам необходимо задавать пару начальных значений, в отличие от обычного варианта функции **reduce**. Одно значение из этой пары является начальным

ПРАКТИКУМ

значением всей операции свертки, другое же является «нулем» операции свертки; оно необходимо для выполнения операции свертки в подгруппах. Так, например, если операцией свертки является операция суммирования чисел, то «нулем» будет число 0, а если операцией свертки будет операция конкатенации строк, «нулем» операции будет пустая строка. В общем случае, что будет являться «нулем», для операции свертки не вычислить. Поэтому пользователь нашей многозадачной версии функции `reduce` должен передавать как начальное значение операции свертки, так и «нуль» этой операции. В нашей реализации мы ожидаем, что эти два значения будут переданы в виде пары (кортежа) `{InitValue, PortionInitValue}`, где `InitValue` – начальное значение всей операции свертки, `PortionInitValue` – «нуль» операции свертки.

Теперь посмотрим на то, чем же занимаются у нас рабочие процессы (т.е. на функцию, выполняемую рабочими процессами). Это функция `portion_worker/5`, определенная в модуле `parallel_reduce` (но не экспортируемая из этого модуля):

```
portion_worker(Fun, SourcePortion, InitValue, Index, MasterPid) ->
  AggrValue = lists:foldl(Fun, InitValue, SourcePortion),
  MasterPid ! {result, Index, AggrValue}.
```

Как мы уже говорили ранее, рабочие процессы у нас «одноразовые», т.е. выполняют свою задачу и заканчивают работу. Именно поэтому функция, которую выполняют рабочие процессы, настолько проста: в ней мы вычисляем значение операции свертки для группы и посылаем сообщение главному процессу с вычисленным значением. Сообщение имеет вид `{result, Index, AggrValue}`, где `Index` – индекс исходной порции данных, `AggrValue` – значение операции свертки для исходной порции. В качестве начального значения `InitValue` для операции свертки порции данных мы передаем «нуль» операции свертки (при создании рабочих процессов).

Пришла пора проверить, что наша многозадачная версия функции `reduce` работает правильно. Для этого компилируем соответствующие модули и запускаем консоль среды выполнения языка Erlang. Так как у нас функция `parallel_reduce:portion_reduce/4` имеет три варианта, все эти три варианта было бы неплохо проверить. Для начала в качестве операции свертки возьмем операцию сложения чисел. Вызов `parallel_reduce:portion_reduce(fun(Item, Agg) -> Item + Agg end, [], {3, 0}, 5)` вернет нам в качестве значения число 3, т.е. начальное значение операции свертки. Вызов `parallel_reduce:portion_reduce(fun(Item, Agg) -> Item + Agg end, [1, 2, 3], {3, 0}, 5)` вернет число 9. Это сумма всех чисел из списка `[1, 2, 3]` с величиной 3 в качестве начального значения суммы. Так как количество элементов в списке – 3, а размер порции – 5, то сумма всех чисел вычисляется при помощи второго варианта функции `parallel_reduce:portion_reduce/4`, т.е. не многозадачным способом. Выполняем вызов `parallel_reduce:portion_reduce(fun(Item, Agg) -> Item + Agg end, [1, 2, 3, 4, 5, 6], {3, 0}, 2)` и получаем в результате значение 24. Легко проверить, что сумма всех чисел из списка с начальным значением этой суммы 3 будет равна 24. В этом вызове размер порции равен 2, а количество элементов в списке – 6; так что при вызове будут созданы 3 рабочих процесса.

Предыдущая операция свертки была коммутативной операцией; давайте проверим работу нашей многозадачной версии

Арность функции

Арность функции – это количество аргументов, которые необходимо передать функции во время ее вызова. В языке Erlang арность функции обозначается следующим образом: `func_name/func_arity`, где `func_name` – имя функции, а `func_arity` – арность функции. Так, например, `some_func/0` обозначает функцию, арность которой 0 (это означает, что при ее вызове не надо передавать никаких аргументов), а `some_func/3` – функцию, арность которой 3 (это означает, что при вызове функции

необходимо передать 3 аргумента). Так как Erlang – язык динамический (то есть типы переменных определяются в нем во время выполнения), то перегрузка некоторой функции возможна, только если функции с одним и тем же именем (определенные в одном модуле) имеют разное количество аргументов, т.е. арность. Следует также добавить, что некоторая функция может состоять из нескольких вариантов, что тоже в своем роде некоторая перегрузка этой функции.

функции `reduce` в случае, когда операция свертки коммутативной не является. В качестве такой некоммутативной операции свертки возьмем операцию конкатенации строк. Как и в предыдущем случае, необходимо проверить работу всех трех вариантов функции `parallel_reduce:portion_reduce/4`. Вызов `parallel_reduce:portion_reduce(fun(Item, Agg) -> Agg ++ Item end, [], {"++", ""}, 2)` вернет нам в качестве значения строку «++», т.е. начальное значение операции конкатенации. Результатом выполнения вызова `parallel_reduce:portion_reduce(fun(Item, Agg) -> Agg ++ Item end, [{"aa", "bb"}], {"++", ""}, 5)` будет строка «++aabb». Легко проверить, что конкатенация всех строк из исходного списка с начальным значением «++» даст нам строку «++aabb». При этом количество элементов в списке равно 2, а размер порции равен 5; это означает, что результат конкатенации строк мы получили при помощи второго варианта функции `parallel_reduce:portion_reduce/4`. И, наконец, результатом выполнения вызова `parallel_reduce:portion_reduce(fun(Item, Agg) -> Agg ++ Item end, [{"aa", "bb", "cc", "dd", "ee", "ff"}], {"++", ""}, 2)` будет строка «++aabbccddeeff». Очевидно, что эта строка является результатом конкатенации всех строк из исходного списка, с начальным значением «++». Так как размер списка – 6 элементов, а размер порции данных – 2, то этот вызов будет обработан третьим вариантом функции `parallel_reduce:portion_reduce/4`; при этом будет создано 3 рабочих потока.

Теперь создадим многозадачную версию функции `map`, рабочие процессы которой будут обрабатывать не отдельные элементы исходного списка, а порции из элементов. Как и раньше, у нас будет один главный процесс и несколько вспомогательных рабочих процессов. В главном процессе (в котором мы инициируем выполнение нашей версии функции `map`) мы будем разбивать данные на порции, создавать необходимое количество рабочих процессов и заданий для них, собирать результаты работы от всех рабочих процессов и, наконец, преобразовывать собранные промежуточные результаты в итоговый список. При этом рабочие процессы у нас продолжают быть «одноразовыми», т.е. они выполняют свою задачу, отсылают результаты работы главному процессу и заканчивают свое существование в этом бренном

»

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

мире. Поэтому мы можем не останавливаться подробно на принципах работы, а сразу посмотреть код соответствующих методов. Рабочая функция главного процесса (она же точка входа в многозадачную версию функции `map`) `portion_pmap/3` определена в модуле `parallel_map` (и, естественно, объявлена экспортируемой функцией из этого модуля):

```
portion_pmap(_Fun, [], _PortionSize) -> [];
portion_pmap(Fun, SourceList, PortionSize)
when length(SourceList) <= PortionSize ->
lists:map(Fun, SourceList);
portion_pmap(Fun, SourceList, PortionSize) ->
process_flag(trap_exit, true),
MasterPid = self(),
PortionCount = parallel_common:calc_portion_
count(length(SourceList), PortionSize),
PreparedData = parallel_common:prepare_data(PortionSize,
SourceList),
lists:foreach(fun({Index, Portion}) -> spawn_link(fun() ->
portion_worker(Fun, Portion, Index, MasterPid) end) end,
PreparedData),
EmptyStorage = array:new([size, PortionCount], {fixed, true},
{default, none}),
FullStorage = parallel_common:collect_result(EmptyStorage,
PortionCount),
process_flag(trap_exit, false),
lists:append(array:to_list(FullStorage)).
```

Как и в случае функции `parallel_reduce:portion_reduce/4`, мы определяем три варианта функции. Первый обрабатывает случай пустого списка; второй – случай, когда количество элементов в списке меньше размера порции (в этом случае нет смысла распараллеливать задачу); и, наконец, третий – общий случай.

Рассмотрим функцию, которую будут выполнять рабочие процессы. Это функция `portion_worker/4`, определенная в модуле `parallel_map` (но не экспортируемая из него):

```
portion_worker(Fun, SourcePortion, Index, MasterPid) ->
DestPortion = lists:map(Fun, SourcePortion),
MasterPid ! {result, Index, DestPortion}.
```

Так как рабочие процессы у нас опять же «одноразовые», то выполняемая ими функция очень проста: мы вычисляем результат операции отображения порции и возвращаем его главному процессу при помощи посылки сообщения вида `{result, Index, DestPortion}`, где `Index` – индекс исходной порции данных, `DestPortion` – результат применения операции отображения к исходной порции.

Проверим, что новая многозадачная версия функции `map` работает правильно. Для этого скомпилируем соответствующие модули и запустим консоль среды выполнения языка Erlang. Так как у функции `parallel_map:portion_pmap/3` определено три варианта, необходимо проверить работу их всех. Начнем с первого варианта, обрабатывающего случай пустого исходного списка. Вызов `parallel_map:portion_pmap(fun(Item) -> lists:reverse(Item) end, [], 4)` возвращает пустой список, как и ожидалось. Второй вариант – когда количество элементов в списке меньше размера порции (тогда дополнительных процессов мы не создаем, а все вычисления выполняем в вызывающем процессе). Вызов `parallel_map:portion_pmap(fun(Item) -> lists:reverse(Item) end, ["str13", "str667"], 4)` вернет список `["31rts", "766rts"]`, как и ожидается. Так как список содержит 2 элемента, а размер порции 4, то мы можем быть уверены, что наш вызов будет выполнен вторым вариантом функции `parallel_map:portion_pmap/3`. И, наконец, проверим последний вариант функции `parallel_map:portion_pmap/3`. Вызов `parallel_map:portion_pmap(fun(Item) -> lists:reverse(Item) end, ["str13", "str67", "str667", "str909"], 2)` вернет нам список `["31rts", "76rts", "766rts", "909rts"]`. Очевидно, что это значение является правильным. При последнем вызове размер порции – 2, а исходный список содержит 4 элемента; то есть создадутся два рабочих потока для выполнения операции отображения каждой порции данных.

Давайте внимательно посмотрим на созданные нами выше многозадачные версии функций `map` и `reduce` и сравним, как эти методы реализуют свою функциональность (сравнивать, что эти методы делают, очевидно, не имеет смысла). В обоих случаях у нас есть главный процесс (он же процесс, в котором инициировался вызов функции) и несколько вспомогательных «одноразовых» рабочих процессов. В обоих случаях в главном процессе мы разбиваем данные на порции, создаем нужное количество рабочих процессов и заданий для них, собираем результаты работы от всех рабочих процессов и, наконец, преобразовываем собранные промежуточные результаты в итоговый результат. А в рабочих процессах, опять же в обоих случаях, мы выполняем задание, отсылаем результат работы обратно главному процессу, и все.

Видно, что оба эти метода организуют свою работу одинаковым образом. Они отличаются только функцией, обрабатывающей порции данных в рабочих процессах, и функцией, преобразовывающей промежуточные результаты в итоговый результат. В реализации функции `map` для обработки порции данных в рабочем процессе используется функция `lists:map/2`, а для преобразования промежуточных результатов в итоговый результат –

Зачем нужен append

Почему мы используем `list:append/1` для объединения промежуточных результатов в итоговый результат в методе `parallel_map:portion_pmap/3` (как и в методе `parallel_map:portion_gen_pmap/3`)?

Порция исходных данных (которую обрабатывает рабочий процесс) является списком, после обработки порции рабочим процессом мы также получаем список (т. к. мы реализовываем многозадачную версию операции отображения), поэтому коллекция промежуточных результатов (а это у нас массив) хранит списки в качестве своих элементов. Функция `array:to_list/1` возвращает список элементов, которые хранились в массиве; в нашем случае это будет список, элементами которого будут списки. Соответственно, возникает задача о пре-

образовании такого «глубокого» списка в плоский список. В общем случае эта задача решается при помощи функции `lists:flatten/1`, но в нашем случае (когда у нас список, элементами которого являются списки из элементов) эту задачу также можно решить при помощи функции `lists:append/1`.

Возникает вопрос: почему мы не использовали для решения этой задачи более очевидную функцию `lists:flatten/1`, а использовали функцию `lists:append/1`? Ответ на этот вопрос достаточно неожиданный: все дело в строках. Как мы помним (см. [LXF149](#)), строки не являются отдельным типом данных; вместо этого строки являются списками, элементами которых являются символы (или целые положительные числа, что то же самое). Поэтому,

если элементами являются строки, то результат выполнения функции `lists:flatten/1` даст совсем не тот результат, который ожидается. Так, например, результатом выражения `lists:flatten(["aa", "bb"], ["cc"])` будет строка «aabbcc», тогда как мы ожидали список из строк `["aa", "bb", "cc"]`. Именно поэтому мы и используем функцию `lists:append/1`: она не имеет подобных побочных эффектов при работе со строками. Мораль всего этого следующая: необходимо с осторожностью подходить к использованию некоторых функций для работы со списками (таких как `lists:flatten/1`), если элементами списка могут быть строки, т. к. вы можете получить не тот результат, на который рассчитывали.

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

функция `lists:append/1`. В реализации функции `reduce` в обоих случаях применяется функция `lists:foldl/3`, только с разными параметрами. Логично спросить: а зачем мы сейчас сравниваем наши реализации функций `map` и `reduce`? Ответ, пожалуй, очевиден: во-первых, мы хотим избежать дублирования кода в уже существующих реализациях методов `map` и `reduce`, а во-вторых, хотим в дальнейшем создавать параллельные версии еще каких-либо методов с минимальными затратами. В нашем случае видно, что мы можем выделить общую часть в реализации методов `parallel_map:portion_pmap/3` и `parallel_reduce:portion_reduce/4`. Все выделяемые общие методы мы помещаем в модуль `parallel_portion_helper` (и, соответственно, в файл `parallel_portion_helper.erl`).

Начнем с функции, которую выполняют рабочие процессы. Для создания ее общей версии достаточно заменить конкретные функции, используемые для обработки порции в рабочих процессах на некоторую функцию, передаваемую как параметр:

```
portion_worker(Fun, SourcePortion, Index, MasterPid) ->
  DestPortion = Fun(SourcePortion),
  MasterPid ! (result, Index, DestPortion).
```

Здесь параметр `Fun` и является той самой функцией, которая обрабатывает входную порцию данных. Следует заметить, что эта функция не экспортируется из модуля `parallel_portion_helper`. Теперь выделим общую часть из рабочих функций главных процессов (и, соответственно, точек входа) для наших реализаций методов `map` и `reduce`. Как уже говорилось, рабочие функции главных процессов отличаются только следующими аспектами: функцией, которая обрабатывает порции данных в рабочих процессах, и функцией, которая преобразовывает промежуточные результаты в итоговый результат. Очевидно, что общая часть должна содержать обе эти функции в качестве параметров. С учетом всего сказанного, общая часть рабочей функции главного потока должна выглядеть следующим образом (естественно, что эта функция является экспортируемой из модуля `parallel_portion_helper`):

```
portion_core(WorkerFun, FinalAggrFun, SourceList, PortionSize) ->
  process_flag(trap_exit, true),
  MasterPid = self(),
  PortionCount = parallel_common:calc_portion_count(
    length(SourceList), PortionSize),
  PreparedData = parallel_common:prepare_data(
    PortionSize, SourceList),
  lists:foreach(fun({Index, Portion}) -> spawn_link(fun() ->
    portion_worker(WorkerFun, Portion, Index, MasterPid) end,
    PreparedData),
  EmptyStorage = array:new([size, PortionCount], {fixed, true},
    {default, none}),
  FullStorage = parallel_common:collect_result(
    EmptyStorage, PortionCount),
  process_flag(trap_exit, false),
  FinalAggrFun(array:to_list(FullStorage)).
```

Здесь параметр `WorkerFun` является функцией, которая применяется для обработки порций исходных данных в рабочих процессах, а параметр `FinalAggrFun` – функция для преобразования промежуточных результатов в итоговый результат. Помимо этих двух параметров, эта функция принимает также исходный список `SourceList` и размер порции `PortionSize`. Может возникнуть вопрос: можем ли мы вынести в метод `parallel_portion_helper:portion_core/4` дополнительные варианты, которые были в методах `parallel_map:portion_pmap/3` и `parallel_reduce:portion_reduce/4` и предназначались для обработки ситуаций, когда исходный список либо пуст, либо его размер не больше размера порции? Для ответа сравним варианты функций `parallel_map:portion_pmap/3` и `parallel_reduce:portion_reduce/4`, которые обрабатывают ситуацию пустого списка с данными. Функция `parallel_map:portion_pmap/3` в этом случае возвращает пустой список, а функция `parallel_reduce:portion_reduce/4` – начальное зна-

чение операции свертки. Видно, что для этого варианта возвращаемые значения специфичны. Точно так же можно увидеть специфику и в вариантах функций `parallel_map:portion_pmap/3` и `parallel_reduce:portion_reduce/4`, которые обрабатывают ситуацию, когда размер списка не превышает размера порции. Именно по этой причине таких вариантов нет в функции `parallel_portion_helper:portion_core/4`.

Теперь посмотрим, как можно переписать функции `parallel_map:portion_pmap/3` и `parallel_reduce:portion_reduce/4`, используя созданную функцию `parallel_portion_helper:portion_core/4`. Новые (переписанные) функции будут называться `parallel_map:portion_gen_pmap/3` и `parallel_reduce:portion_gen_reduce/4`, соответственно. Реализация `parallel_map:portion_gen_pmap/3` функции выглядит следующим образом:

```
portion_gen_pmap(_Fun, [], _PortionSize) -> [];
portion_gen_pmap(Fun, SourceList, PortionSize)
when length(SourceList) =< PortionSize ->
  lists:map(Fun, SourceList);
portion_gen_pmap(Fun, SourceList, PortionSize) ->
  WorkerFun = fun(SourcePortion) -> lists:map(Fun,
  SourcePortion) end,
  parallel_portion_helper:portion_core(WorkerFun, fun
  lists:append/1, SourceList, PortionSize).
```

Здесь функция `WorkerFun` создается на основе функции `lists:map/2`, а функцией `FinalAggrFun` является функция `lists:append/1`. Аналогичным образом выглядит реализация функции `parallel_reduce:portion_gen_reduce/4`:

```
portion_gen_reduce(_Fun, [], {InitValue, _PortionInitValue}, _
  PortionSize) -> InitValue;
portion_gen_reduce(Fun, SourceList, {InitValue, _
  PortionInitValue}, PortionSize)
when length(SourceList) =< PortionSize ->
  lists:foldl(Fun, InitValue, SourceList);
portion_gen_reduce(Fun, SourceList, {InitValue, PortionInitValue},
  PortionSize) ->
  ReduceFun = fun(List) -> lists:foldl(Fun, InitValue, List) end,
  PortionReduceFun = fun(List) -> lists:foldl(Fun, PortionInitValue,
  List) end,
  parallel_portion_helper:portion_core(PortionReduceFun,
  ReduceFun, SourceList, PortionSize).
```

Здесь и функция `WorkerFun`, и функция `FinalAggrFun` создаются на основе функции `lists:foldl/3`; главное отличие между этими построениями в том, что в качестве параметра `Acc0` (начальное значение) в первом случае берется «ноль» операции свертки, а во втором – начальное значение операции свертки.

Для тестирования работы функций `parallel_map:portion_gen_pmap/3` и `parallel_reduce:portion_gen_reduce/4` мы можем использовать те же сценарии, что и для тестирования функций `parallel_map:portion_pmap/3` и `parallel_reduce:portion_reduce/4`. Поэтому дублировать результаты тестирования функций `parallel_map:portion_gen_pmap/3` и `parallel_reduce:portion_gen_reduce/4` мы здесь не будем (желающие могут провести его сами и убедиться, что все работает, как надо).

Сегодня мы сделали очередной шаг в нашем практикуме: создали многозадачные версии функций `map` и `reduce`, которые обрабатывают порции исходных данных параллельно. Настраивая размер порции, мы можем управлять производительностью наших функций (мы об этом поговорим в одном из следующих номеров). Более того, мы смогли найти и выделить общее ядро у соответствующих версий функций `map` и `reduce`. Важность этого результата в том, что при создании всех дальнейших многозадачных версий функций `map` и `reduce` мы будем создавать для них некоторое общее ядро. Это поможет избежать дублирования кода и облегчит создание многозадачных версий каких-либо еще функций (но об этом мы поговорим в следующий раз). **LXF**

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 Монтирование Android
- 2 Яркость монитора
- 3 Проблемы с разбиением диска
- 4 Разрешения монитора
- 5 Настройку раздела подкачки
- 6 Перенаправление почты в Kmail

1 Переброс файлов с Android

В Я купил планшет Nexus 7 на Android и очень им доволен. Увы, без «но» не обошлось: я не могу смонтировать его как USB-устройство. Когда я подключаю по USB свой телефон с Android, на телефоне появляется окно, где он монтируется как накопитель, но на планшете, похоже, ничего этого нет.

Конечно, файлы можно перебросить через Dropbox, но у меня несколько ГБ музыки и видео. Почему они отказались от такой вполне приличной системы? И как мне перебросить файлы?

Грант [Grant]

Проблема с подключением накопителей по USB в том, что в каждый момент времени их можно подключить только к одной операционной системе. Это означает, что прежде чем смонтировать ее на компьютере, ее нужно размонтировать на телефоне. Поэтому на телефоне появляется предупреждение о том, что нужно закрыть приложения, использующие SD-карту. На устройствах без съемных носителей, вроде Nexus 7, все еще хуже: размонтировать файловую систему, используемую операционной системой, нельзя. Поэтому в новых устройствах для решения этой проблемы используется MTP (Media Transfer Protocol – протокол передачи мультимедиа). MTP был разработан Microsoft и используется для передачи файлов на медиа-проигрыватели и обратно, точно так же, как PTP (Picture Transfer Protocol – протокол передачи изображений) используется для передачи изображений с цифровых камер. Существует несколько решений для монтирования устройств MTP в Linux; с устройствами на Android лучше всего работает `jmtarfs` (<http://research.jacquette.com/jmtarfs-exchanging-files-between-android-devices-and-linux/>). Это файловая система FUSE, которая использует библиотеку `libmtp` (<http://libmtp.sourceforge.net>). Последняя должна быть в репозиториях вашего дистрибутива, но, возможно, придется установить `jmtarfs` из исходников. Есть еще `mtpfs`, но сейчас она работает не так хорошо, как `jmtarfs`. Впрочем, обе они – проекты новые, и все может измениться.

После установки список всех подключенных устройств с поддержкой MTP можно получить командой

```
jmtarfs -l
```

Чтобы смонтировать первое найденное устройство – обычно в каждый момент времени у вас подключено только одно устройство – укажите точку монтирования после команды `jmtarfs`. Этот каталог должен существовать и иметь право на запись для пользователя, от имени которого запущена команда

```
jmtarfs ~/nexus
```

Вы увидите сообщение об отсутствии файла `.mtpz-data`, но на него можно не обращать внимания: файл используется только для аутентификации Microsoft. Для размонтирования устройства используется команда

```
fusermount -u ~/nexus
```

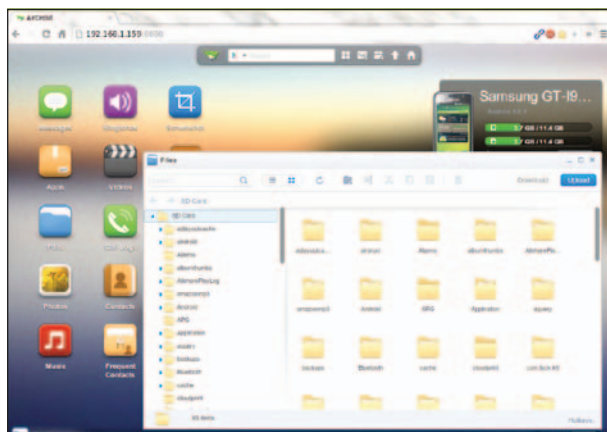
Это означает, что устройство можно смонтировать, но исключительно вручную. Можно сделать и так, чтобы устройство автоматически монтировалось `udev`, создав для этого два файла. Первый – правило `udev`; назовите файл `/etc/udev/rules.d/90-mtpmount.rules`. Число имеет значение, так как это правило должно запускаться позже, чем правила `libmtp`. В файле должна быть одна строка:

```
ATTR{idVendor}=="18d1",ATTR{idProduct}=="4e42",  
RUN="/usr/local/bin/mountnexus.sh &"
```

Идентификаторы производителя и продукта указаны для Nexus 7 – для других устройств их покажет `lsusb`. Монтирование длится до нескольких секунд, а `udev` не любит программы, которые не завершаются сразу, поэтому параметр `RUN` указывает путь к скрипту (который может быть любым), который делает всю работу. Создайте файл скрипта с командами:

```
#!/bin/sh  
mkdir -p /media/nexus7  
jmtarfs -o allow_other /media/nexus7
```

Запустите скрипт с терминала, чтобы убедиться, что он работает. Параметр `allow_other` необходим потому, что `udev` запускает скрипты от имени суперпользователя-`root`, а вам нужна возможность записывать в каталог как обычный пользователь. Для работы скрипта может потребоваться добавить эту опцию в `/etc/fuse.conf`. Затем подключите планшет и откройте каталог `/media/nexus7` в любимом файловом менеджере.



Вместо монтирования устройства на Android можно установить на нем AirDroid и общаться с ним через браузер – но это не слишком удобно для синхронизации музыкальной коллекции.

2 Непрямое перенаправление

В Яркость на моем компьютере регулируемая, и в Fedora есть несколько связанных с этим файлов. При входе в систему яркость максимальна, и на основном экране это режет глаза. Я зашел в программу управления видеокарты и нашел настройки монитора, но их изменения сохраняются только в текущем сеансе. Стоит выйти из системы и войти снова, и яркость снова максимальная.

Я подумал: «Если это происходит при каждом входе в систему, я найду нужный файл настройки и пропишу его в своем `.bashrc`». Это файл `/sys/class/backlight/acpi_video0/brightness`, содержащий число от 0 до 9, и его владелец – `root`. Я решил добавить следующую команду в `.bashrc`:

```
echo 3 > /sys/class/backlight/acpi_video0/  
brightness
```

Команда не сработала, так как я не суперпользователь. Я попробовал запустить ее с командной строки с `sudo`, но это тоже не сработало. Однако я могу открыть другое окно терминала, войти в систему командой `su` – и скопировать эту строку в новый терминал; тогда яркость меняется.

Родж Парментер [Rog Parmenter]

Яркость устанавливается командой `xbacklight`, которая не требует прав `root`, но работает не во всех системах, поэтому мы опишем общий принцип, так как он пригодится не только для этой задачи. Перенаправление вывода с `sudo` часто вызывает путаницу из-за способа интерпретации команд оболочкой. Мы читаем слева направо, первым видим `sudo` и предполагаем,

что все последующее будет выполняться от имени root. Но оболочке ближе принцип Йоды [персонаж «Звездных войн», который постоянно переставлял слова местами, – прим. пер.], и она сперва реализует перенаправление, так что команда

```
sudo somecommand >somefile
```

запускает команду от имени root, но пишет в файл от имени обычного пользователя. Решение – воспользоваться **tee**, командой, которая перенаправляет все со стандартного ввода в стандартный вывод и в файл:

```
echo 3 | sudo tee /sys/class/backlight/acpi_video0/brightness
```

Теперь **echo** выполняется от имени обычного пользователя, а запись в файл – от имени root. **3** появится и на терминале, и с целью от этого /dev/null:

```
echo 3 | sudo tee /sys/class/backlight/acpi_video0/brightness > /dev/null
```

Проблема с запуском **sudo** из **.bashrc** еще не снята, так как команда остановится и запросит пароль. Однако можно настроить запуск команд от имени root и без запроса пароля. Откройте файл **/etc/sudoers** – лучше делайте это с помощью **visudo**, так как она проверяет синтаксис и позволяет избежать неприятных ситуаций из-за неправильной конфигурации в **sudoers**:

```
sudo visudo
```

Нам ни к чему, чтобы кто попало мог запустить **tee** от имени root, ведь это фактически дает доступ на запись к любому файлу в системе, и нужно указать всю команду полностью, следующим образом:

```
username ALL = NOPASSWD: /usr/bin/tee /sys/class/backlight/acpi_video0/brightness
```

Это позволяет одному указанному пользователю запустить данную конкретную команду от имени root без запроса пароля. Путь до команды указывается весь целиком, иначе можно будет создать файл с таким же именем в другом месте

и получить доступ root. Сохраните файл и проверьте его командой

```
echo 3 | sudo tee /sys/class/backlight/acpi_video0/brightness
```

Она должна уменьшить яркость без запроса пароля. Если команда работает, добавьте ее в свой **.bashrc** или, еще того лучше, поместите в скрипт, запускаемый автоматически при загрузке рабочего стола.

3 Home-мания

В Я пользуюсь своим нетбуком Asus EeePC 901 в основном для заходов в Интернет, но хочу применять его и для других целей, типа игр и тому подобного. Сейчас на нем установлен Crunchbang Linux; это легкий и приятный дистрибутив, и в нем было довольно простое руководство по настройке разделов специально для EeePC; там сама операционная система устанавливается на раздел размером 4 ГБ, а еще один раздел размером 16 ГБ используется как /home.

И вот такая у меня проблема: при каждой установке какой-то программы через менеджер пакетов она всегда попадает на раздел с системой, а не с домашним каталогом. В результате довольно большой раздел в 16 ГБ пустует, а системный раздел забивается программами.

Возможно, я не понял чего-то простого, но как заставить менеджер пакетов устанавливать программы в каталог /home?

Чез Джельф [Chaz Gelf]

О У меня был 901, и эти два диска мне тоже досадили. Я схитрил и объединил их в один с LVM, но делать это нужно до установки дистрибутива. Информация о том, куда копируются файлы пакетов, находится в самих пакетах, и ее нельзя изменить. Если пакет хочет поместить их в /usr, они обязаны отправиться в /usr. Это высечено на камне; зато на камне не высечено, где должен быть сам /usr. Выполните

следующую команду, чтобы узнать, какой каталог занимает больше всего места:

```
sudo du -hx --max-depth=1 /
```

Решив, какой каталог вы хотите переместить на другой диск (хорошие кандидаты – /var и /usr), перезагрузите нетбук в live-дистрибутив, чтобы не возиться с корневой файловой системой в работающей ОС. Теперь смонтируйте два диска, проще всего – из терминала, командой

```
mkdir -p /mnt/{root,home}
```

```
mount /dev/sda1 /mnt/root
```

```
mount /dev/sdb1 /mnt/home
```

Затем переместите весь каталог **usr** в файловую систему и создайте на ее месте пустой каталог:

```
mv /mnt/root/usr /mnt/home/
```

```
mkdir /mnt/root/usr
```

Наконец, добавьте строку в файл **/etc/fstab**, которая свяжет точку монтирования **usr** с новым каталогом:

```
/home/usr /usr none defaults,bind 0 0
```

Эта строка должна идти после строки с монтированием /home. Прodelать такое можно с /usr или /var, либо с обоими каталогами, но не пытайтесь затрагивать другие каталоги корневой файловой системы. Такие каталоги, как /etc, /lib, /bin и /proc, должны оставаться в корневой системе.

Можно было бы переразбить второй диск, создать раздел для /usr и смонтировать его в **fstab** – пуристы скажут, что так и надо было сделать – но наш вариант проще. Важно и то, что он дает нам больше гибкости в распределении дискового пространства на диске в 16 ГБ, ведь его и так немного.

Есть еще один вариант, но для него потребуются переустановка системы. Это изначальная установка системы на диск большего размера. У вас может быть корневая файловая система размером 16 ГБ с разделом /home и раздел подкачки размером 4 ГБ, а остальное монтируется как дополнительное хранилище. »



Коротко про...

Initramfs

З аглянув в каталог /boot или файл меню загрузчика, вы увидите ссылки на файлы initrd. Это виртуальные диски образов ramfs. Виртуальный диск, как следует из названия – хранилище данных, которое выглядит как диск и проживает в оперативной памяти, а файл initrd – это виртуальный диск, сохраненный в файл. Таково определение, но зачем это нужно?

Initrd (виртуальный диск инициализации – initial ramdisk) – это виртуальный диск, загружаемый ядром при его запуске. Он становится корневой файловой системой, и с него запускаются скрипты настройки системы, после чего управление передается настоящему корнево-

му разделу на жестком диске. Основная функция виртуального диска – загрузка модулей ядра. Дистрибутивы предназначены для работы на самом разном оборудовании, а это означает, что большинство модулей драйверов являются встроенными.

Будь они вкомпилированы в образ ядра, у нас бы получилось огромное ядро, медленно загружаемое и на 90 % избыточное. Поэтому в Linux предусмотрены загружаемые модули ядра; но модули ядра, необходимые для монтирования корневой файловой системы, из нее самой загрузить нельзя, и нам нужен способ предварительной загрузки драйверов контроллера жесткого диска, файловой системы и, возможно,

некоторых других, таких как LVM или dm-crypt. Это и делает виртуальный диск; скрипт **linuxrc** на нем загружает соответствующие модули, запускает все необходимые программы установки (например, для LVM или для зашифрованной корневой файловой системы) и затем переключает ее на корневой раздел на жестком диске.

Образ initrd – это файловая система в файле **cpio**, упакованном **gzip**, так что можно распаковать, смонтировать, изменить и упаковать ее следующими командами:

```
cd /mnt/tmp
```

```
zcat /path/to/initrd | cpio -id
```

```
#modify files here
```

```
find . -depth | cpio -o | gzip >/path/to/newinitrd
```

4 Капризный монитор

Из дистрибутивов сентябрьского DVD только в Slack Puppy монитор настроился правильно. Все дистрибутивы устанавливая разрешение моего монитора (на самом деле, телевизора с диагональю 37 дюймов) примерно в 1360×728, но при этом текст во всех приложениях читается плохо.

Только при разрешении 1280×720 текст и иконки получаются четкими и с верными пропорциями, но зато рабочий стол сдвинут влево. Но при смене разрешения на более низкое Slack Puppy отображает текст и иконки нормально и заполняет всю площадь экрана телевизора.

Кстати, два других «щенка», Wary и Lucid, отлично ладят с моим монитором, а Mascup — нет. Утилита позиционирования KDE не помогает.

Как Slack Puppy удается то, что не удается другим? Можно ли что-то с этим сделать? Сейчас я пользуюсь Ubuntu 10.04, и никаких проблем с монитором нет.

Питер Стивенсон [Peter Stevenson]

Похоже, в большинстве дистрибутивов неправильно определяются характеристики монитора и поэтому задается неверный режим. Неверное разрешение на ЖК-или светодиодном мониторе выдает именно те текстовые артефакты, о которых вы говорите. В отличие от старых ЭЛТ-мониторов, ЖК-мониторы состоят из дискретных пикселей, и при попытке задать другое разрешение драйвер попытается отобразить точку там, где ее физически нет.

Одно изменение разрешения не снимает проблемы, так как по-прежнему неверны другие аспекты видеорежима. Есть два способа решения проблемы. Первый — загрузить спецификации своего монитора, покорпеть над map-страницей `xorg.conf` и пробовать разные настройки до тех пор, пока результат вас не удовлетворит.

Второй — схитрить и стянуть конфигурацию из Slack Puppy. Для этого загрузитесь в Slack Puppy и задайте нужные настройки монитора. Эти настройки будут записаны в файл `/etc/X11/xorg.conf` — скопируйте его на флешку или на раздел на жестком диске. Теперь перезагрузитесь в другой дистрибутив, скопируйте этот файл обратно в `/etc/X11` (вам понадобится доступ пользователя `root` или `sudo`) и перезапустите X одной из следующих команд, в зависимости от своего дистрибутива, или перезагрузив компьютер.

```
/etc/init.d/dm restart
/etc/init.d/xdm restart
/etc/init.d/gdm restart
/etc/init.d/lightdm restart
```

Немного странно, что правильно распознал монитор только один дистрибутив, но так как по сути это телевизор, он может сообщать свои настройки с ошибками, что и заставляет другие дистрибутивы выбирать неверные настройки.

5 Раздел подкачки

Я устанавливаю Crunchbang в VirtualBox и выделил 11,4 ГБ для виртуального жесткого диска. При разбивке диска на разделы с помощью мастера мне не удается выделить много места в корневой раздел или раздел подкачки — большая часть диска выделяется в `/home`. Поскольку я не собирался хранить на нем много файлов, мне захотелось разбить диск по-другому, выделив около 8 ГБ на корневой раздел, 1 ГБ на раздел подкачки, а остальное — в `/home`. И все вроде прекрасно, но утилита разбиения не дает мне возможности сделать раздел разделом подкачки.

Heiowge, с форумов

И в установщике Crunchbang 10, и в установщике Crunchbang 11, которые мы пробовали, возможность создать раздел

подкачки имелась. Однако и в работающей системе достаточно легко создавать, изменять и выделять дисковое пространство в разделы подкачки. Самый простой способ обойти проблему с этим установщиком — создать раздел нужного размера, но не задавать его тип и не монтировать его (если тип нужно задать обязательно, выберите его произвольно). После установки и загрузки системы преобразуйте этот раздел в раздел подкачки — либо в `GParted`, либо вручную, запустив в терминале команду `mkswap`:

```
sudo mkswap /dev/sdN
Затем добавьте его в /etc/fstab в строке
/dev/sdN none swap sw 0 0
```

Если в вашем `fstab` для определения дисков используются UUID и вы намерены этого придерживаться, то уникальный идентификатор раздела подкачки можно узнать командой

```
sudo blkid /dev/sdN
```

Для разделов подкачки можно задать и метки файловых систем, как и для «обычных» файловых систем, чтобы конфигурация в `fstab` была более понятной. Для этого создайте раздел подкачки командой

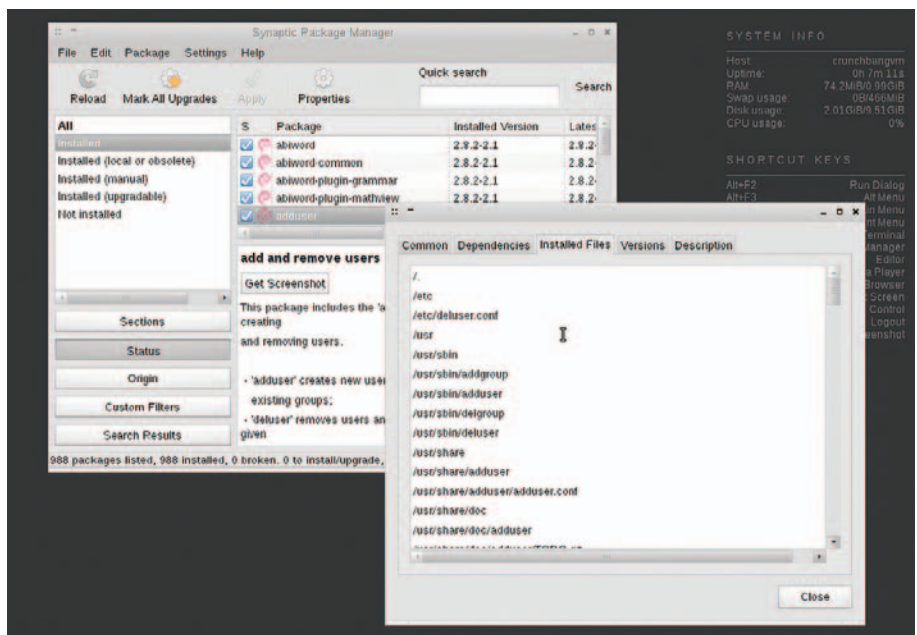
```
sudo mkswap --label myswap /dev/sdN
и добавьте его в /etc/fstab строкой
LABEL=myswap none swap sw 0 0
```

6 Перенаправление почты в KMail

Недавно я установил Kubuntu 12.04, чтобы лучше узнать KDE. Я всегда был и остаюсь пользователем Ubuntu. Надеюсь, Грэм, наш постоянный эксперт по KDE, поможет мне с проблемой, которая выглядит проблемой зеленого новичка. Похоже, `KMail 4.8.4` не умеет перенаправлять почту без включения всех исходных получателей и отправителей. В `Thunderbird` я всегда удалял эти данные, так как они просто отнимают место, к тому же являясь приватными.

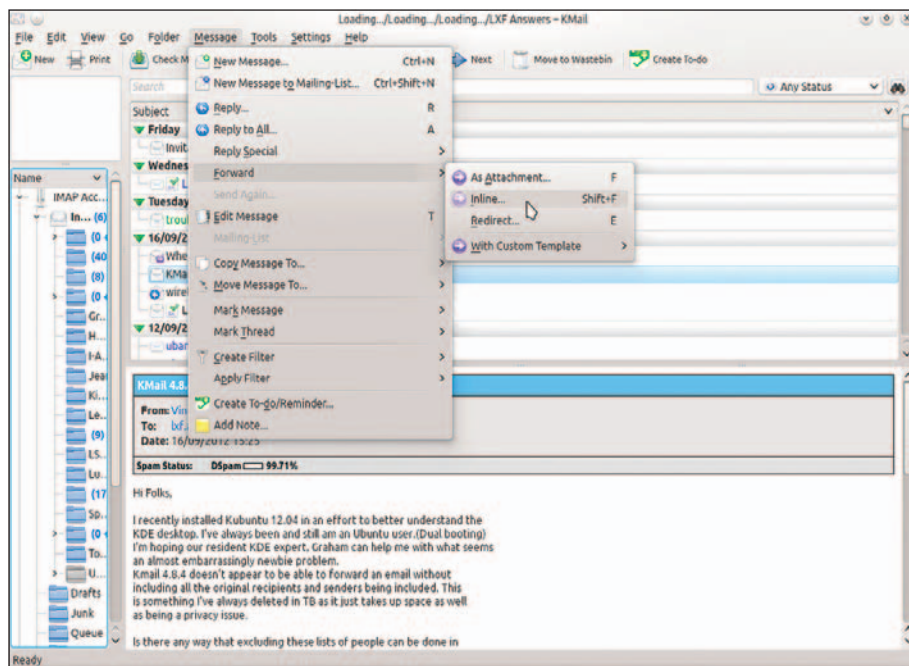
Можно ли исключить эти списки в `KMail`?
Винс [Vince]

KDE у нас любит не только Грэм. Я и сам им пользуюсь с версий 1.x, хотя, признаюсь, откладывал переход с 3.5 на 4.x дольше, чем многие другие, пока последняя не стала по-настоящему удобной. `KMail` поддерживает три способа переправки письма дальнейшим получателям, все они доступны в подменю Forward [Пересылка] меню Messages [Сообщения] или в контекстном меню, которое открывается при щелчке правой кнопкой мыши по сообщению. При переключении [Redirecting] сообщение пересылается как есть, без указания, что оно пересылается вами. Затем есть два варианта перенаправления. В первом случае письмо пересылается как вложение, и его содержимое сохраняется, и часто именно это и нужно, особенно в сфере бизнеса, поскольку тогда понятно, что содержимое пересылаемого письма не изменилось. Наверное, именно этот вариант и используете вы, так как в нем нельзя изменять сообщение. Если выбрать второй вариант, Forward > Inline [Пересылка > Вложенное], то в `KMail` и других почтовых клиентах письмо



При установке файлов из пакетов `.deb` или `.rpm` в них жестко зашито местоположение файлов, но каталог монтирования места назначения можно изменить.

Помогите нам помочь вам



➤ KMail предлагает несколько вариантов перенаправления. Выберите подходящий именно вам.

вставляется в новое сообщение с пометкой «Forwarded Message» в теме письма.

Перед отправкой можно отредактировать письмо, удалив другие адреса (не говоря уже о нескольких корпоративных подписях, вложен-

ных друг в друга). В *Thunderbird* этот метод используется по умолчанию; все что вам нужно сделать в *KMail* – поработать над своей мышечной памятью и нажимать для перенаправления письма Shift+F вместо F. **LXF**

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
```



Часто задаваемые вопросы

Пользователи и суперпользователи

➤ Я только что впервые установил Linux. Зачем понадобилось создавать двух пользователей?

Одним из них был root, или суперпользователь, он используется только для администрирования. Второй – обычный пользователь для обычной работы в системе.

➤ Зачем нужны два пользователя? Нельзя ли администрировать систему от имени своего пользователя?

В Linux есть права доступа к файлам, определяющие, кто является владельцем каждого файла или каталога и кто может читать или записывать в него. У каждого пользователя есть свой домашний каталог, в котором он может делать все что угодно, создавая и удаляя файлы в свое удовольствие. За пределами домашнего каталога каждый

пользователь может читать файлы и запускать программы, но не изменять или удалять их. Это обеспечивает безопасность и защищенность. Вы не можете случайно удалить системные файлы, и даже запуск вложения из вредоносного письма не испортит систему, потому что у вас нет нужных прав доступа. Административный доступ нужен, например, для установки программы. Вы переключаетесь на пользователя root, выполняете нужные действия и переключаетесь обратно.

➤ То есть при каждой установке программы или других подобных действиях мне придется выходить из системы и заходить в нее как пользователь root? Это не слишком весело.

Нет, выходить из системы не придется, как не придется и заходить в графическую среду от имени root (в некоторых дистрибутивах этого вообще сделать нельзя). От имени root нужно лишь запустить отдельную программу. Многие программы уже так и настроены. Если вы попытаетесь запустить утилиту настройки системы YaST в SUSE или Центр управления Mandriva [Mandriva Control Centre], у вас спросят пароль root. Для любой другой программы откройте терминал, наберите **su**, введите пароль и затем запустите программу, набрав ее имя.

➤ Компьютером пользуюсь я один. Нужно ли на самом деле каждый раз заходить в систему?

В большинстве дистрибутивов можно настроить автоматический вход

в систему как конкретный пользователь (не root!). Это удобно, если вы единственный пользователь. Если вы пользуетесь KDE, это можно настроить в Менеджере входа в систему [System Administration > Login Manager] в Центре управления KDE [KDE Control Centre]. Пользователи Gnome должны выбрать пункт меню Desktop > System Settings > Login Screen [Рабочий стол > Настройки системы > Окно входа в систему] в главном меню и включить автоматический вход в систему [Automatic Login] на вкладке General [Общее]. Однако дважды подумайте, прежде чем включить эту опцию на ноутбуке. Если вы его потеряете или его украдут, то каждый, кто включит его, получит беспрепятственный доступ к вашей почте и другой приватной информации.



LXF Hot Picks



Майк Сондерс

Издав самые недостижимые и укромные уголки Интернета, Майк точно знает, где скрываются лучшие образчики открытого кода.

MKVToolNix » Glances » Tasque » Midori » o42a » Zathura » cclive » Trigger Rally » Quabro » John the Ripper » Lazarus

Программа для создания видеофайлов

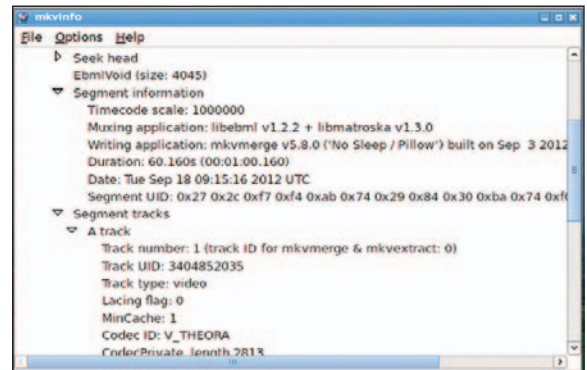
MKVToolNix

Версия 5.8.0 Сайт www.bunkus.org/videotools/mkvtoolnix/

Видео в Интернет – это сущая солянка сборная. Практически с начала века нам приходилось сражаться с самыми разными видами всевозможных кодеков и форматов, и даже просто для того, чтобы обеспечить достойный уровень просмотра, приходится устанавливать кучу модулей расширения. Ситуация немного сдвинулась в лучшую сторону благодаря широкому распространению Flash, но он все равно оставался постоянно летающим плагином с кучей ошибок, а не идеальным решением для просмотра видео. И хотя сейчас у нас в качестве «стандартного» формата видео принят

H.264, вокруг него кипят страсти по поводу нарушения патентов.

И, как обычно, сообщество свободного ПО предложило решение – или, на самом деле, решения. Одно из них – Matroska, полностью открытый формат, то есть возможность хранить аудио- и видеопотоки в единственном файле, подобно AVI в мире Windows. Обычно файлы Matroska име-



Используя `mkvinfo-gtk -g`, за которым следует имя файла, вы можете получить подробную информацию о файле Matroska.

«С начала века нам приходилось сражаться с кодеками.»

ют расширение `.mkv`, и их можно просматривать во многих плеерах и в некоторых браузерах. Matroska стремится занять место стандартного формата мультимедиа в сети, чтобы мы все могли реализовать на своих сайтах видео, не думая о проблемах с патентами и авторскими отчислениями. Для создания файлов Matroska требуется набор инструментов; вот тут-то и появляется *MKVToolNix* (мы очень бегло рассматривали его в **LXF141**, но с тех пор он ушел далеко вперед). Это набор утилит командной строки и графических утилит для получения информации из файлов `.mkv`, извлечения из них данных или объединения с другими источниками. Самая полезная – *mkvmerge*, доступная в своем воплощении для GUI через *mmg*. После запуска перед вами появляется несколько простоватый, но по большей части очевидный интерфейс.

Чтобы приступить к конвертированию видео, нажмите на кнопку Add в панели Input и выберите существующее видео. Затем ниже вы увидите треки, содержащиеся в файле (видео и аудио), и если их выбрать, откроются опции в панели со вкладками внизу. Удовлетворившись выбором, нажмите на Start mixing внизу, и вы увидите журнал, отображающий результаты. Затем можете загрузить видео на свой сайт и помочь сделать Сеть, которую мы все так любим, свободной и открытой для будущего.

Исследуем интерфейс MKVToolNix

Опции

В File > Options настраивается поведение интерфейса.

Global

В этой вкладке можно разделять полученные файлы и делать их WebM-совместимыми.

Источники

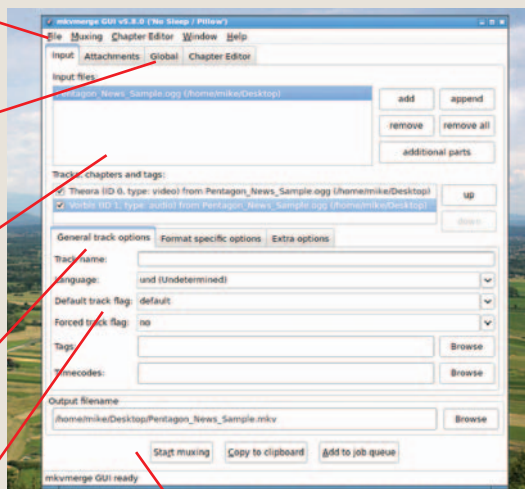
Именно сюда вы добавляете существующее видео для конвертирования в формат `.mkv`.

Содержание

Загрузив файл источника, здесь вы сможете увидеть компоненты аудио и видео.

Настройки результата

Изменяйте настройки полученного файла, например, коэффициент сжатия, в Format specific [Указание формата].



Конвертирование

Выберите имя файла для полученного файла и жмите Add to job queue [Добавить в очередь задач], чтобы создать партии для конвертирования.

Системный монитор

Glances

Версия 1.4.1. Сайт <http://bit.ly/QWBRnN>

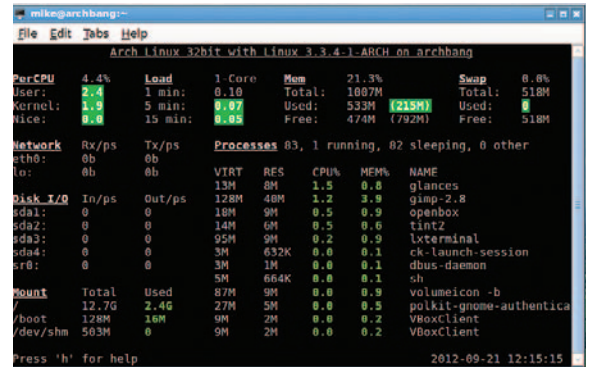
Работа системного администратора никогда не заканчивается. Приходится отслеживать самые последние проблемы безопасности, обновлять пакеты, возиться с файлами настроек и исправлять то, что напортили пользователи. И даже когда все работает нормально, надо приглядывать за компьютерами, чтобы у них не исчерпалось ОЗУ или место на жестком диске. Благодаря архитектуре UNIX и открытости Linux, для администраторов не составляет особого труда заглянуть вовнутрь системы, однако брести на ощупь в /proc и жонглировать целой связкой инструментов командной строки не очень-то элегантно.

Glances – это программа терминала, предоставляющая мгновенную информацию о системных ресурсах. Она старается объединить на одном экране все, что может потребоваться системному администратору. Экранный снимок объясняет это лучше всего; на нем вы видите то, что отображается в Glances по умол-

чанию, демонстрируя нагрузку на CPU, потребление памяти и подкачки, активность сети, I/O диска, точки монтирования и список процессов. Хотя вся информация собрана в одну кучу (и принимая во внимание тот факт, что он предназначен для работы на терминалах 80×24 – т.е., машинах без X), Glances прост для понимания благодаря использованию атрибутов терминала, таких как цвет и жирный шрифт. И цвет здесь отнюдь не визуальное украшение: нет, зеленый фон под статистическими данными может переокраситься в случае ухудшения ситуации.

В более крупных терминалах список процессов расширяется, отображая большее количество столбцов, добавляя, например, PID и User, но в остальном оста-

«Жонглировать инструментами командной строки незелегантно.»



Если царит умиротворяющий зеленый цвет, повода для паники нет. Пока нет.

ется таким же. Есть также ограниченная возможность индивидуальной настройки, с помощью которой вы можете изменить последовательность списка процессов (например, по использованию CPU или RAM), а с помощью флага командной строки `-t` можно изменить частоту обновления экрана.

И, наконец, Glances – превосходная программа для работы на SSH-соединении с сервером или на любой машине, за которой вы хотите пристально понаблюдать.

Менеджер задач

Tasque

Версия 0.1.11 Сайт <https://live.gnome.org/Tasque>

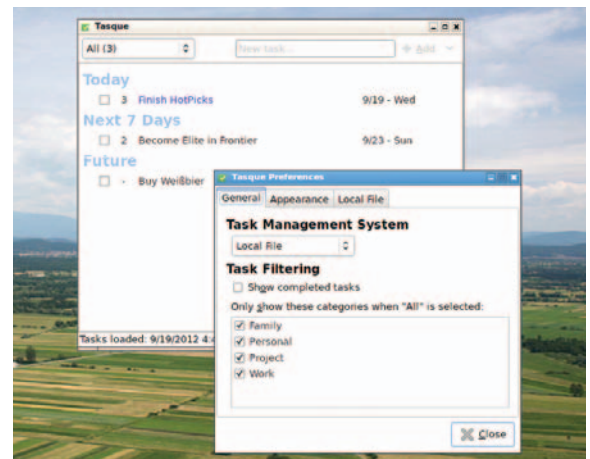
Загляните в мозги автора HotPicks, и вы увидите множество наклеек-напоминаний: программы, с которыми надо познакомиться, программы, рекомендованные читателями, программы, о которых мы уже рассказывали, программы, которые не скомпилируются, ну и так далее. Мы не напрашиваемся на сочувствие, но всегда радуемся, если одна из программ, на которые мы наткаемся, может облегчить жизнь.

Tasque – менеджер задач. Их в мире Linux несметное количество, практически для любого набора инструментов и языка, но в Tasque нас подкупила его простота. Он не пытается перегрузить свой интерфейс всякими излишествами и не страдает от заразной болезни под названием скеоморфизм – то есть от попыток заставить программу выглядеть, как «реальный» объект, добавляя в качестве фона эффекты деревянных панелей или фотографии ручек.

При самом первом запуске Tasque предоставляет на выбор две модели хранения ваших списков: либо интеграцию с замечательным сайтом www.rememberthemilk.com, либо хранение данных локально. Затем предлагается практически пустой список для добавления ваших задач. Просто введите имя в поле сверху справа и щелкните по кнопке Add [Добавить]. Затем можете настроить задачу, выбрав нужную дату, категорию и приоритет для нее.

Для завершения задачи надо просто поставить галочку в поле. Чтобы вывести диалоговое окно Preferences, щелкните правой кнопкой по значку с галочкой на рабочем столе или на системном лотке менеджера окон (у вас нет возможности

«Он нам понравился потому, что мы сразу его применили.»



Tasque не является волшебным зельем, исцеляющим от проволочек, это просто хорошо сделанное небольшое приложение.

получить к нему доступ через главное окно). Больше всего нас огорчила ограниченная возможность настройки Tasque – вы не можете изменить количество приоритетов или добавить новые категории; но это и обуславливает его простоту. Он нам настолько понравился потому, что мы сразу же его применили при подготовке данного выпуска HotPicks; а любой инструмент, который с ходу может оказаться полезным в плане увеличения производительности, заслуживает внимания.

Браузер

Midori

Версия 0.4.7 Сайт <http://bit.ly/O4DD81>

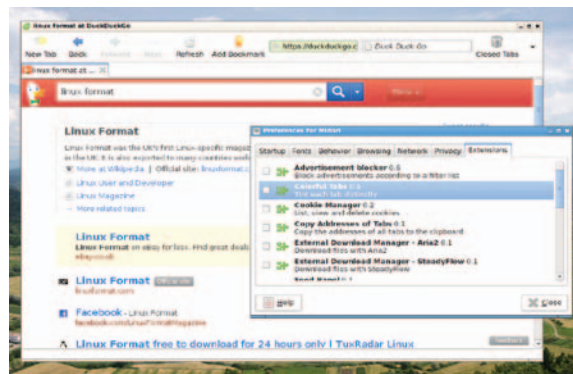
О т функциональных перегрузок, без сомнения, больше всего страдают web-браузеры. Каждый браузер начинает с громких заявлений о своей «быстроте и легкости», но по мере добавления новых функций требования к ОЗУ начинают расти, и тот шустрый маленький браузер, который вы так любили когда-то, становится довольно громоздким.

Firefox является тому превосходной иллюстрацией: он появился на свет как экспериментальное ответвление Mozilla под названием Phoenix, которое избавилось от всякого старого хлама ради того, чтобы обеспечить низкую требовательность к ресурсам. Но вскоре Firefox начал набирать вес, и хотя не так давно он улучшил свои скоростные показатели, все же у него остаются проблемы с управлением памятью.

Для тех, кто трясется над мегабайтами, есть альтернатива в виде Midori. Он берет движок рендеринга WebKit (изначально —

ответвление KHTML из KDE, а теперь применяется в Google Chrome и Safari от Apple) и упаковывает его в простой интерфейс GTK. И он действительно нетребователен к ОЗУ: например, для доступа к домашней странице Midori он использует только 46 МБ ОЗУ. Сравните с Firefox, который потребляет до 115 МБ! А если умножить эту разницу на несколько вкладок, то экономия памяти получится весьма ощутимая.

По части интерфейса Midori вполне оправдывает ваши ожидания и по виду, и по ощущениям, используя те же комбинации клавиш для навигации и управления вкладками, что и Firefox с Chrome. Кнопки на верхней панели напоминают стиль Netscape своим крупным размером, но их можно перенастроить, превратив



» Хотя данный браузер не отличается богатством функций, система расширений может несколько исправить эту ситуацию.

в более мелкие значки или вообще в текст. Очень удобно, что по умолчанию имеется значок с мусорной корзиной, обеспечивающий быстрый доступ к недавно закрытым вкладкам, а поисковая машина настроена на DuckDuckGo.

Midori обладает рядом расширений, таких, как программа блокировки рекламы и жесты мышью, и его система Preferences весьма достойна. В целом нельзя сказать, что Midori опережает соперников на световые годы, но он выполняет свою работу компетентно, аккуратно и с куда меньшими затратами памяти, чем Firefox. Особенно хорошо Midori работает в Xfce, и если вы планируете вдохнуть с помощью Linux новую жизнь в старенький компьютер, он должен стать вашим главным выбором.

«Интерфейс Midori вполне оправдывает ваши ожидания.»

Язык программирования

o42a

Версия 0.2.3 Сайт <http://o42a.org>

М ногие языки программирования пытались слепо копировать свои человеческие эквиваленты, однако результаты отнюдь не впечатляли. Возьмите, например, COBOL:

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO-WORLD.
PROCEDURE DIVISION.
DISPLAY 'Hello, world'.
STOP RUN.
```

Не особо привлекательно, да? Однако COBOL — язык древних, так что новые попытки, как, например, o42a, всегда заслуживают внимания.

По словам разработчиков, программа o42a должна выглядеть «ближе к естественному английскому тексту, чем программа, написанная на C-подобном языке программирования». Это компилированный, статически типизированный, сильно декларативный язык с примесью императивных функций программирования.

Версия вышеприведенного кода в o42a выглядит так:

```
Use namespace 'Console'.
@Main := * {
Print "Hello, World!" nl.
}
```

Это пример отметки конца утверждения точкой, что сильно напоминает использование ; в C-подобных языках. Наблюдаются и другие попытки сделать язык человекоподобным, ну, например, как эта:

```
i < 10? ...
```

Это означает: «если величина переменной i меньше десяти, вернуться к началу текущего цикла». Объектная модель o42a основана на прототипах, и объекты про-



» Документация понятна и сопровождается множеством примеров — дает высший балл.

исходят непосредственно от других объектов, без классов и интерфейсов. Объект 'void' является предшественником всех объектов, а поля внутри объектов доступны с помощью двоеточий, например: «foo: bar». Язык пока на ранней стадии разработки, и разработчики уделяют основное внимание правильности синтаксиса и структуры, так что на данный момент есть только прототип компилятора на Java (с LLVM в качестве генератора кода движка).

Но, невзирая на ранний номер серии 0.2.x, в нем уже имеется солидный объем документации.

«Невзирая на номер 0.2.x, в нем солидный объем документации.»

Программа просмотра документов

Zathura

Версия 0.2.1 Сайт <http://pwmt.org/projects/zathura>

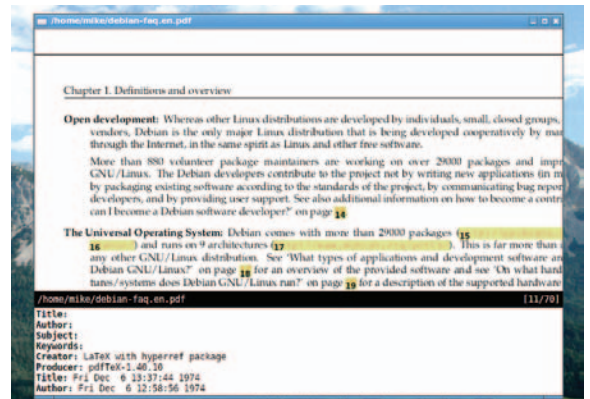
Мы знаем, о чем вы думаете: «Опять программа для просмотра документов? Разве их в Linux уже не прорва?» И вы будете правы — это категория программ, заезженная практически до смерти. Мы говорим «практически», потому что, обнаружив *Zathura*, мы поняли, что о ней стоит рассказать. В то время, как большинство программ для просмотра документов работают по принципу «наведи-щелкни», что удобно для неопытных пользователей, *Zathura* предназначена для более продвинутых пользователей, которые в основном предпочитают клавиатуру.

Ее интерфейс отличается минимализмом: загрузите, например, файл PDF, и простительно будет предположить, что перед вами одно из старых приложений X11, типа *xman* или *xedit*. Здесь нет кнопок или меню — вы просто видите строку состояния внизу экрана, где отображаются имя файла и номер страницы. Теперь можете осуществлять навигацию по доку-

менту с помощью обычного курсора и клавиш Page Up/Down, но *Zathura* также поддерживает *Vi*-подобные функциональные клавиши, так что можете проматывать текст с помощью клавиш h, j, k, и l (используйте заглавные J и K, чтобы проматывать сразу целые страницы). И *Vi*-образное управление этим не ограничивается — у вас есть разные команды, например, **:bmark** для закладок, **:print** для печати и **:info** для отображения технических подробностей документа. Вы даже можете использовать классическую **:q** от *Vi* для выхода.

Еще одна полезная клавиша — **f**: если нажать ее, она выделит внешние и внутренние ссылки в документе, поставив перед ними номера. И если вы просматриваете документ, у которого есть

«Отдых от необходимости хвататься за мышшь вернул наши улыбки.»



» Вот так выглядит экран после нажатия **f** для выделения ссылок документа.

содержание, вам, например, не нужно тянуться за мышью — просто нажмите на **f**, а затем на соответствующий номер нужной вам ссылки. В нее включено также множество других опций и команд, согласно описанию в справочнике (**man zathura**), и хотя программа поддерживает лишь файлы PDF, PostScript и DjVu, возможно, в будущем будут доступны и другие форматы через систему модулей расширения. Мы взяли *Zathura* как программу для просмотра PDF по умолчанию, и через пару дней отдых от необходимости хвататься за мышшь при просмотре документов — особенно технических, со множеством ссылок — вернул на наши нечеловечески милые лица прежние улыбки.

Программа для скачивания видео

cclive

Версия 0.7.10 Сайт <http://cclive.sourceforge.net>

Внимание, фанаты YouTube: сколько раз вы смотрели чудесные видео с котятми, падающими со стола, или с Линусом Торвальдсом, который, не стесняясь в выражениях, высказывает свое мнение о Nvidia, и страстно хотели сохранить его на свой жесткий диск? Есть множество модулей расширения браузеров для просмотра YouTube HTML и извлечения оттуда видеоконтента для прямого скачивания, но они часто неудобны в использовании и периодически ломаются. Разве не чудесно было бы обзавестись альтернативой командной строки?

Что ж, к счастью, она существует, и именуется *cclive*. Этот разумный инструмент берет YouTube URL и извлекает видеоданные в отдельный файл:

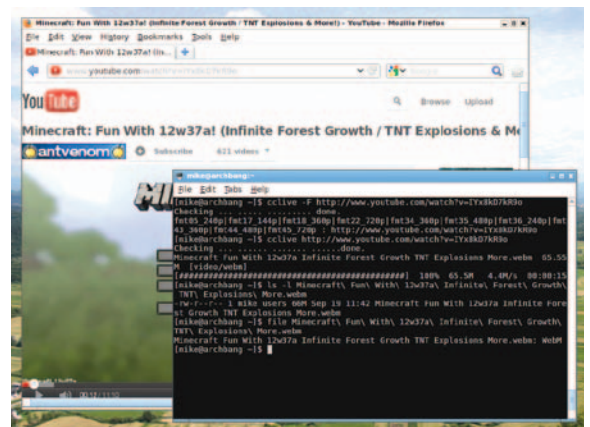
```
cclive http://www.youtube.com/watch?v=IYx8kD7kR9o
```

Это дает нам файл, получившие свое название по названию видео, в формате WebM. Далее, форматы, доступные

для каждого видео, можно определить с помощью флага **-F**, как видно на экранном снимке; есть разрешения от 240 до 720. Затем вы можете скомандовать *cclive* скачать определенный формат, используя опцию **--format**. Если посреди процесса скачивания видео с вашим интернет-соединением случился облом, вы сможете возобновить скачивание с помощью **-c**, и дать результату разные имена файлов с помощью **-o**.

cclive включает также продвинутые функции, как способность маскироваться под определенный web-браузер с помощью опции **--agent**. Можете также воспользоваться **--throttle** для ограничения скорости загрузки и загрузить его

«cclive также поддерживает сайты вроде Vimeo и Dailymotion.»



» Познакомьтесь со списком имеющихся форматов для видео, и затем скачайте видео — проще некуда.

через прокси-сервер (это удобно, если видео недоступно в вашей стране). При помощи опции **--exe** можно исполнять дополнительные команды в файле сразу после загрузки, что весьма удобно, если вам нужно сделать конвертирование в другой формат или извлечь аудио.

Помимо YouTube, *cclive* также поддерживает сайты вроде Vimeo и Dailymotion. Существовал проект по созданию для него графической оболочки (*Abby* на <http://code.google.com/p/abby>), но он уже некоторое время не работает. Будем надеяться, что какой-нибудь предприимчивый хакер *Qt* его воскресит...

HotGames Развлекательные приложения

Гоночная игра

Trigger Rally

Версия 0.6.0 Сайт <http://bit.ly/PWaeMB>

Со времен расцвета *Sega Rally* мы тоскуем по гонкам. Видите ли, обычные имитаторы вождения очень хороши, но уж больно незатейливы: вы крутите руль влево – машина едет влево. Крутите вправо – машина едет вправо. И так далее. А вот в гоночных играх вы тратите половину своего времени на огибание поворотов, пытаетесь направить машину в сторону, противоположную той, куда ее заносит. Это волшебное чувство, и при обнаружении *Trigger Rally* для Linux нам показалось, будто все наши рождественские праздники наступили разом.

Точнее, часть праздников, поскольку есть только версия 0.6.0. Как бы то ни было, она уже в хорошей форме – есть достойный физический движок и приличный выбор автомобилей. Что касается соответствия законам гравитации – оно вызывает некое подозрение, потому

что машины ведут себя так, словно они на Марсе, но это лишь добавляет веселья. Здесь есть три машины и 33 трассы, от реалистичных джунглей и каньонов до безумных вулканов и скачков по лунной поверхности.

Сейчас здесь есть только режим одиночной игры, делящийся на три опции: практика, события и одиночные гонки. В каждом случае вы едете на время, прокладывая маршрут по трассе так, чтобы ваша машина прошла все красные обручи (см. экранный снимок). Можете управлять машиной мышью или клавиатурой – в обоих случаях нужно двигаться ровно и с филигранной точностью, чтобы спра-

«Нужно двигаться ровно, чтобы справиться на скорости.»



► Прыгающие красные обручи сильно отличаются от привычных контрольных точек.

виться с машиной на скорости, иначе перевернетесь.

Trigger Rally уже доставляет массу удовольствия, и вы ощущаете глубокое удовлетворение, если проходите трассу на солидной скорости, не врезаясь в какую-нибудь часть пейзажа. Правда, ощущаешь себя довольно одиноко – наличие других машин, по нашему ощущению, придало бы игре некую завершенность. Кстати, если вы компилируете из исходника, загляните в `doc/README.txt`, потому что система компиляции несколько нестандартна.

Игра в мяч битой

Quabro

Версия 0.6.1 Сайт <http://sourceforge.net/projects/quabro/>

Вряд ли швыряние мяча по кирпичам битой может считаться особо впечатляющим рецептом игры, однако *Quabro* придал ему интересный поворот. С точки зрения игрового процесса, это практически стандартный арканоид: мяч скачет по экрану, и с помощью управляемого вами устройства внизу экрана надо направить мяч так, чтобы он ударил по всем кирпичам (и разбил их). Добавок надо ухитриться, чтобы мяч не пролетел мимо вашей биты и не обрушился вниз.

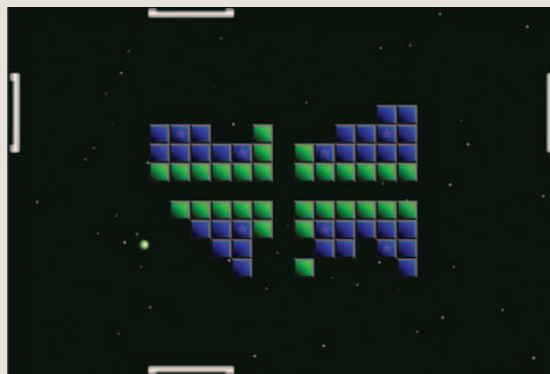
Но при том, что традиционные арканоиды позволяют управлять одной битой, *Quabro* дает вам четыре. Горизонтальные – впрочем, так же, как и вертикальные – синхронизированы, и в конечном итоге вам просто надо использовать стрелки управления влево/вправо и вверх/вниз. Но зато теперь,

когда мяч может выскочить с любой стороны экрана, игра становится значительно более трудной.

Еще у вас есть бонусы. Определенные блоки скрывают в себе определенные плюсы, которые увеличивают набранные вами очки и лопасть биты или сужают ее. Вы можете заставить мяч крутиться, сдвинув биту в момент соприкосновения с мячом, и мяч будет по-разному реагировать на края биты.

Итак, надо на многое обращать внимание: собирать полезные бонусы (избегая вредных), следя при этом, чтобы мяч не вылетал за границы экрана, и развора-

«С точки зрения процесса это стандартный арканоид.»



► Блоки со звездами предоставляют хорошие или плохие бонусы, прямо как в жизни.

чивать мяч так, чтобы он попал по определенному блоку.

Ф-фу. Однако от нее на удивление трудно оторваться, а спокойная космическая музыка и забавные звуковые эффекты создают очень приятную атмосферу. Самое приятное, что все уровни (их 20) хранятся в текстовых файлах, и вам будет нетрудно создать собственный и поиграть с необычными образцами.

Уровни сложности в игре влияют на скорость мяча – самый простой сильно тормозит, и на то, чтобы очистить весь экран, уходит вечность, поэтому, впервые приступая к игре, поставьте средний.

Программа для взлома паролей

John the Ripper

Версия 1.7.9 Сайт www.openwall.com/john/

Нередко, когда дело касается безопасности, самым слабым звеном оказываются пароли. У вас может быть самый бронированный в мире дистрибутив, и вы можете каждую наносекунду скачивать новые заплатки безопасности, но если у вас (или ваших пользователей) легко поддающиеся взлому пароли, они представляют собой зияющий лаз, который так и ждет, когда же им кто-то воспользуется. Итак, как определить, безопасен ли пароль? Длинный пароль из смеси букв и цифр – это определенно неплохо, но можно также оценить его программой взлома паролей.

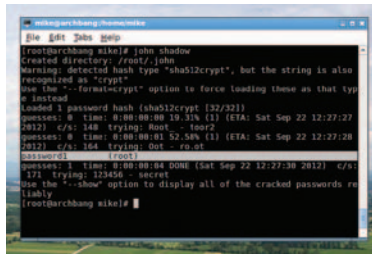
John the Ripper предназначен для «определения слабых паролей Unix» и поддерживает широкий ряд хэшей и шифров. Пользоваться им просто: сделайте копию своего системного файла с паролем (например, `/etc/shadow`) и запустите *John*, следующим образом:

```
sudo cp /etc/shadow .
```

`john shadow`

John начнет искать совпадения в своем словаре (`/usr/share/john/password.lst`) с хэшами в файле паролей. В зависимости от того, сколько пунктов в вашем файле паролей, это может занять от нескольких минут до нескольких часов.

В словаре содержится более 3500 наиболее часто используемых паролей. Если совпадения не обнаружатся, все равно нет гарантии, что ваш пароль идеален, но, по крайней мере, вы убедитесь, что он не относится к часто используемым.



► Наш супернадежный пароль “password1” мигом отправился отдыхать под напором разрушительной мощи *John*.

IDE Pascal

Lazarus

Версия 1.0 Сайт www.lazarus.freepascal.org

Эта IDE Pascal очень долго пыталась дорасти до версии 1.0, учитывая, что серия разработки 0.9 уходит корнями аж в 2004 год. *Lazarus* предоставляет полную среду разработки для программистов Pascal, и если вы много программировали в дни Windows 9x, она вам напомнит *Delphi*. Она построена на *FPC*, *Free Pascal Compiler*, который тоже некоторое время топтался на месте.

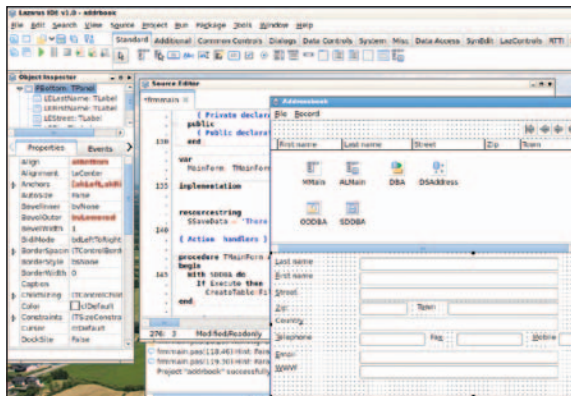
Как IDE, *Lazarus* дает вам все, что нужно: выделение синтаксиса; редактор сворачивания кода; программу разработки форм «наведи-и-щелкни» для создания пользовательского интерфейса; отладки с помощью GDB; и многочисленные шаблоны проектов. Она использует собственную библиотеку *Lazarus Component Library*, среду и набор инструментов, которая позволяет написать один набор кода GUI и заставить его работать на нескольких платформах.

Библиотека статично соединяется с любыми исполняемыми файлами, которые

вы создаете, и вы легко сможете упаковать в пакеты свою работу для разных дистрибутивов – сложных инструкций по установке писать не придется.

Это отлаженная, зрелая IDE, отличный способ начать программировать на Pascal, и хотя эта программа не нова, ее релиз 1.0 – событие, которое достойно упоминания. **LXF**

► *Lazarus* работает на Linux, Windows и Mac OS, и имеет библиотеки для базы данных, сети и разработки игр.



Также вышли

Новые и обновленные программы, тоже достойные внимания...

► Cinelerra 4.4

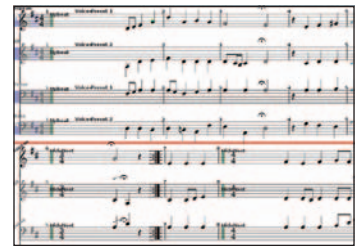
Полнофункциональная среда для производства аудио и видео, с уймой эффектов.
<http://heroinewarrior.com/cinelerra.php>

► OptiPNG 0.7.3

Без потерь сжимает ваши файлы PNG до более мелкого размера.
<http://optipng.sourceforge.net>

► Denemo 0.9.4

Симпатичный редактор музыкальной нотации, на базе набора *LilyPond*.
www.denemo.org/HomePage



► Создайте свой ремикс на песню Ребекки Блэк (Rebecca Black) *Friday*, только с открытым кодом.

► Xplorer 0.10.0

Менеджер файлов; особенно хорош в определении типов файлов по их содержанию.
<http://cxplorer.sourceforge.net>

► YAGAC 0.99

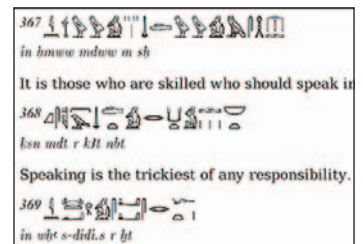
C-программы, определяйте утечки памяти в своих приложениях во время их работы.
<http://freecode.com/projects/yagac>

► ZXTune b2000

Воспроизводит и нарезает музыку ZX Spectrum в широком выборе форматов.
<https://code.google.com/p/zxtune>

► JSesh 6.0

Редактор для древнеегипетских иероглифических текстов. Нет, правда!
<http://jsesh.qenherkhopeshef.org>



► «И сказал ему фараон: птица птица глаз змея человек птица кряк...»

На диске

Пробуйте новую операционную систему уже сегодня!



Некоторые дистрибутивы придерживаются своего графика выхода релизов, некоторые нет, а некоторые его вообще не публикуют. Это означает, что мы не всегда знаем в начале месяца, что будет на диске.

Slackware 14 в какой-то мере застал нас врасплох, но это был приятный сюрприз. Хоть нам и нравится простота в использовании многих дистрибутивов, все же в мире Linux всегда найдется

место подходу, отличающемуся от обычного «наведи-и-щелкни». Мы включили Trisquel, который как-то упустили из вида в начале года, когда был настоящий бум релизов. Это один из немногих дистрибутивов, остающихся на 100% верным принципам свободного ПО.

В этом месяце у нас получился своего рода альтернативный DVD, но ведь суть Linux как раз в том, чтобы являть собой альтернативу — хоть немного.

Зрелость приходит с возрастом

Slackware

Без сомнения, Slackware можно назвать патриархом дистрибутивов Linux. Ну да, до него была парадругая дистрибутивов, но они канули в Лету, а вот Slackware жив и по сей день.

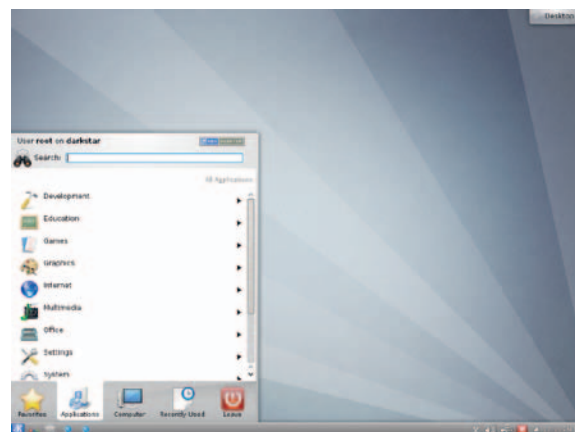
В некотором смысле его можно считать минималистским дистрибутивом. Не потому, что он включает лишь скудный набор программ (это не так) или что он оптимизирован для работы на старых компьютерах (это не так), но потому, что он содержит лишь минимальное количество шелухи поверх программ.

Он не одобряет специфических для каждого дистрибутива инструментов настройки, а менеджер пакетов — это всего

лишь скрипты, необходимые для запуска программ. Пакеты программ как можно более близки к версиям разработчиков.

Если вам нравятся шикарные графические программы установки или супермаркеты приложений, то Slackware, вероятно, не для вас. Однако, если вам нужна система Linux, которая именно этим и является — просто системой Linux, а не тем, чем, по мнению разработчика, система Linux должна являться, то вы явно почувствуете себя в ней как дома.

Версии live на DVD нет, и вам придется установить ее, чтобы с ней познакомиться. Как только загрузитесь с DVD в командную строку, используйте `ctfdisk` для разбиения



Slack использует базовый KDE.

диска на разделы, затем запустите `setup` для установки. Инсталлятор не графический — он работает на базе `curses` — и удивительно беспроblemный. Однако если проблемы все же возникнут, загляните в папку Slackbook на DVD.

Вам не отнять у нас нашей свободы

Trisquel

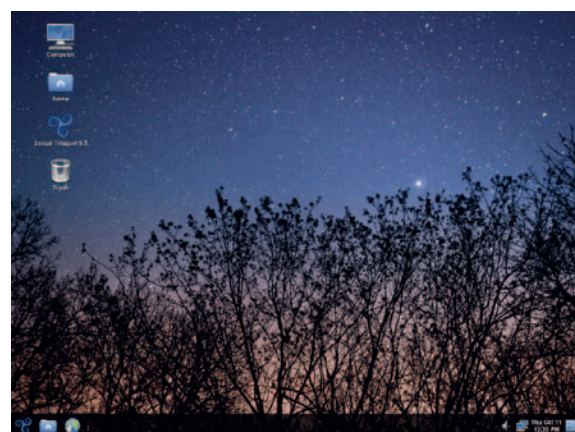
Linux часто говорят, как об ОС с открытым кодом, свободной, как свободно слово. Если отложить в сторону дискуссии о том, должно ли быть в названии GNU/, это не совсем верно. По крайней мере, не всегда. Большинство дистрибутивов Linux в основном открытые, но со включениями ПО с закрытым кодом в тех случаях, когда, в силу разных причин, решение с открытым кодом не оптимально. Очевиднейший пример — драйверы видеокарт, но есть и другие.

Важно это для вас или нет — зависит только от ваших идейных убеждений. Если вас устраивает status quo, то для вас есть масса дистрибутивов. Однако если вы считаете, что свобода ПО — это единственно правильный путь, то ваш выбор окажется довольно ограниченным. Проект Gnu Proj-

ect ведет список действительно свободных дистрибутивов Linux на www.gnu.org/distros/free-distros.html, и мы включили один из них в лице Trisquel.

С технической точки зрения система достойная, но не особо примечательная. Рабочий стол — довольно милая разновидность Gnome 3. Возможно, самый неожиданный выбор среди программ — `ABrowser`. Это ответвление `Firefox`, чья цель — большая свобода: удалены товарные знаки, и он перенаправлен на магазин дополнений с чисто свободным ПО.

«Если вас устраивает status quo, для вас есть масса дистрибутивов.»



Каждый раз, когда вы загружаете Trisquel, Ричард Столлмен улыбается.

Возможно, вы обнаружите, что ваше железо работает с этим дистрибутивом похуже, чем с другими. Дело в том, что в него не включен ни один из двоичных драйверов. Но что значит эффективная работа компьютера по сравнению с истинной свободой?

Не столь крошечный, как Tiny Core, но все-таки

Core Plus

Будем откровенны: между дистрибутивами Linux не такая уж большая разница; может быть, разные менеджеры пакетов, или отлаженная и настроенная индивидуально версия рабочего стола. Однако Tiny Core не похож на большинство дистрибутивов. Вероятно, самое большое отличие в том, что

он запускает все из памяти, а не с жесткого диска. В результате получается поразительно быстрая система, даже на старых компьютерах.

Мы включили в состав диска Core Plus, немного большую по объему версию Tiny Core, хотя странно называть дистрибутив размером 64 МБ большим.



➤ Смотреть здесь особо не на что, но все страсти кипят за сценой.

Несмотря на название, работает с X

AntiX

В поисках дистрибутива для старого оборудования многие автоматически обращаются к Puppy. В этом нет ничего худого, но, поскольку в этом месяце мы придерживаемся альтернативных вариантов, то рассмотрим AntiX.

Он идет с менеджерами окон IceWM и Fluxbox – традиционный выбор для эко-

номии ресурсов. Они и задают тон дистрибутива. Где возможно, в него включены старые приложения, например, XMMS для музыки. Это дает очевидные преимущества в плане производительности и стабильности. К счастью, разработчики не стали доводить это до крайности, и там, где современные программы по-

лучили солидные усовершенствования, они включили их. Например, в нем имеется наш любимый офисный пакет – Libre Office.

«Идет с менеджерами окон IceWM и Fluxbox. Они и задают тон.»

Важно ВНИМАНИЕ!



Прежде чем 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.



➤ Лимон и киви — не самое наше любимое сочетание фруктов, но на ощущения пользователя это никак не влияет.

На диске

Помимо четырех симпатичных дистрибутивов, мы включили в состав диска весь код из учебников, все программы из Notricks и все программы, упомянутые в журнале, рядом с которыми стоит значок «На диске». Если вы установите DVD в компьютер, он должен открыть страницу HTML по умолчанию, но если ваши настройки безопасности не позволяют этого, перейдите в [index.html](#), чтобы открыть для себя все 4,4 Гб прелестей Linux.

Для знакомства с дистрибутивами вложите диск в дисковод и перезагрузите компьютер. Загрузится экран, где вы сможете выбрать то, что вам нужно. Если ваш ПК загружается в обычном режиме, а не с диска, вам придется изменить настройки BIOS на загрузку с DVD. Если вам захочется перенести содержимое диска на устройство USB, мы включили для вас удобный скрипт. Чтобы узнать, как им пользоваться, загляните в [dvd2usb.html](#) на диске. Цифро-

вые подписчики, обратите внимание, что вы можете просто использовать `dd` для перемещения загруженного ISO на устройство USB. Используйте `sudo dd if=lx165.iso of=/dev/sdX` при необходимости поставив путь к загруженному ISO, где X изменяется соответствующим образом для устройства USB. Это уничтожит все данные на диске! Если вы установите не тот диск, он может стереть все данные с вашего 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>LXF162 Октябрь 2012</p> <ul style="list-style-type: none"> » Матч гигантов Какой дистрибутив лучше? » За нашу свободу Лицензии на открытое ПО » SecureBoot Загрузка-то безопасна, но с установкой ясно не все » Скажи пароль А вот "password" в ответ говорить не надо! <p>LXFDVD: Arch, CentOS, Fedora, Mageia и еще 11 дистрибутивов, 9 книг о Linux (на английском языке), горячие новинки и прочее...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_162/ PDF-версия: shop.linuxformat.ru/elxf_162/</p>	 <p>LXF163 Ноябрь 2012</p> <ul style="list-style-type: none"> » Линус Торвалдс Если бы не он, то ничего бы и не было » Галереи изображений Выложите фотки в Сеть в своем стиле » Firefox OS Переносимая надежда мобильной индустрии » Долой бумагу Сохраним леса и снизим пожароопасность офиса <p>LXFDVD: Mint 13 KDE, Raspbian, OpenSUSE 12.2 и еще 2 дистрибутива, 9 книг о Linux (на английском языке), горячие новинки и прочее...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_163/ PDF-версия: shop.linuxformat.ru/elxf_163/</p>	 <p>LXF164 Декабрь 2012</p> <ul style="list-style-type: none"> » Linux ищет бозон Помогаем Большому адронному коллайдеру » Дистрибутивы для детей Как приобщить малышей к Linux » Gnome OS Рабочий стол перекрашивается в операционную систему » Платон, Маркс и... Linux? Философский базис свободного ПО <p>LXFDVD: OpenSUSE, Zorin, Ubuntu и еще 3 дистрибутива, 9 книг о Linux (на английском языке), горячие новинки и прочее...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_164/ PDF-версия: shop.linuxformat.ru/elxf_164/</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



6 месяцев **1230** руб.

12 месяцев **2280** руб.

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: MinStick, менеджер файлов Nemo и многое другое

» Ядро 3.5 » Mate 1.4 » Gimp 1.6 » LibreOffice 3.6 и Python 3
С рабочим столом Gnome или Mate, 32- и 64-разрядные сборки

Linux Mint 14



Январь 2013
LXF DVD 165/166

LINUX
FORMAT

А ТАКЖЕ: Lazarus, Midori, MKVToolnix и многое другое

И ЕЩЕ: Antix 12 » CorePlus 4.7 » Trisquel 5.5

Обновленная версия старейшего дистрибутива

Slackware 14.0

Slackware
14.0

Январь 2013
LXF DVD 165/166

LINUX
FORMAT

Содержание

LINUX FORMAT

Сторона 1

ДИСТРИБУТИВЫ

Anix 12 Облегченный Live CD с рабочими столами IceWM и Fluxbox, 32-разрядная сборка
CorePlus 4.7 Минималистский дистрибутив для слабого «железа», 32-разрядная сборка (ISO-образ)
Slackware 14.0 Обновленная версия старейшего дистрибутива, 32-разрядная сборка (загрузка с **LXF DVD**)
Trisquel 5.5 Дистрибутив на базе Debian, свободный от проприетарных кодов, 32-разрядная сборка (загрузка с **LXF DVD**)

Сторона 2

ДИСТРИБУТИВЫ

Linux Mint 14 Установочные DVD (ISO-образы)
С рабочим столом Mate 32- и 64-разрядные сборки
С рабочим столом Cinnamon 32- и 64-разрядные сборки
ПРОГРАММЫ
WiringPi Реализация библиотеки Wiring для Raspberry Pi
ДОКУМЕНТАЦИЯ: 9 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)
Bash Scripting Подробное руководство по программированию на Bash
Bourne Shell Scripting Начальное руководство по программированию на Bash
Cathedral Bazaar Классический текст Эрика Раммонда [Eric S.Raymond] «Сбор и базар»
The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian
Intro to Linux Начальное руководство по Linux

Linux Dictionary Словарь Linux, объясняющий специфическую терминологию
Linux Kernel in a Nutshell Описание ядра Linux, созданное одним из его выдающихся разработчиков — Греггом Крау-Хартманом [Greg Kroah-Hartman]
System Administrator's Guide Руководство по базовому администрированию Linux
GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

НОТРИСКС

CCLive 0.710 Программа загрузки Flash-видео для просмотра офлайн
Glances 1.4.21 Средство мониторинга системы, написанное на curses

JohnTheRipper 1.7.9 Популярная программа для взлома паролей
Lazarus 1.0 IDE для программирования на Free Pascal
Midori 0.4.7 Облегченный web-браузер на основе WebKit
MKVToolNix 5.8.0 Набор утилит для работы с мультимедийными файлами формата MKV
0424 0.2.3 Система программирования на языке, максимально приближенном к естественному английскому
Quapro 0.6.1 Игра в стиле Sokoban
Tasque 0.111 Простое средство управления заданиями (TODO list)
TriggerRally 0.6.0 Игра, имитатор автомобильных гонок
Zarathrua 0.2.1 Программа для просмотра документов различных форматов

Пожалуйста, перед использованием данного диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

Двеятные диски В маловероятном случае обнаружения дефектов на данной диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех старых платформах, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аппаритурный 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, предоставленных нашей программой или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, скачайте программу для загрузки с нашего сайта: 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. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке – например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле `/etc/default/cdrecord`. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor=0,3,0 12 16M
```

Первое слово в этой строке – 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.



ITmozo.ru

**БОЛЕЕ
4000**
ИТ-ВАКАНСИЙ

Работа в интереснейших стартапах
или крупнейших ИТ-компаниях.

Большой выбор задач, зарплат и
условий



Аппаратно-программный комплекс

DR.WEB OFFICE SHIELD

Комплексное решение задачи антивирусной
и антиспам-защиты для малых и средних предприятий



88 679 руб.

DR.WEB OFFICE
SHIELD TWISTER
Рассчитан на 250 ПК



65 240 руб.

DR.WEB OFFICE
SHIELD NEO
Рассчитан на 50 ПК




WWW.LINUXCENTER.RU/SHOP/ANTIVIR/DR_WEB/

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



*Отдел
дистрибьюции
ГНУ/Линуксцентра
приглашает
дилеров
и дистрибьюторов
к сотрудничеству!*

**Широкая сеть
представительств
в разных городах
позволит вам
оптимизировать
процессы логистики
и доставки товара.**

Подробнее о партнерской программе:
www.linuxcenter.ru/partner/

allbuntu.ru



сообщество
ПОЛЬЗОВАТЕЛЕЙ
UBUNTU



Unix Education Center
Россия, 190000, Санкт-Петербург
Черноморский переулок, дом 4
Тел.: + 7 (812) 611-1575












Футболки GNU/Linux

БОЛЕЕ 100 ВИДОВ. ВСЕ РАЗМЕРЫ И ЦВЕТА



WWW.LINUXCENTER.RU/SHOP/GIFTS/



В февральском номере

Да здравствует Linux Mint!

Mint победил. Но как? Мы оглянулись назад и забежали вперед, и получили инсайдерскую информацию от изнуренного трудами Клемана Лефевра, чтобы разобраться.

Игры – это здорово

Мы послали своего «казачка» в Humble Indie Bundle. Он вышел оттуда с широкой ухмылкой на лице.

Построено на Django

Продолжаем совершенствовать web-программирование в рабочей среде на базе Python.

Снова о хакерах кракерах

Защитите свое свеженькое web-приложение, изучив пути его взлома. Браконьеры против лесников и все такое...

Содержание будущих выпусков может меняться – вдруг нас заблокируют в нашем рождествоупорном бункере.

LINUX FORMAT

Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № Ф077-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж 3000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Елена Ессяк, Светлана Кривошеина, Валентин Развозжаев, Тихон Тарнавский, 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 «Лепень»

188330, Ленинградская обл., Гатчинский р-н, пос. Сиверский, Вокзальная ул., 4
Заказ 7455

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Грэм Моррисон (Graham Morrison) graham.morrison@futurenet.com

Заместитель редактора Эндрю Грегори (Andrew Gregory)

agregory@futurenet.com

Редактор диска Бен Эверард (Ben Everard) ben_everard@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоса

(Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.com

Выпускающий редактор Гэри Уокер (Gary Walker) gary.walker@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Джоно Бэкон (Jonno Bacon), Нейл Ботвик (Neil Bothwick), Крис Браун (Chris Brown), Бен Эверард (Ben Everard), Джульетта Кемп (Juliet Kemp), Джон Лэйн (John Lane), Грэм Моррисон (Graham Morrison), Майк Сондерс (Mike Saunders), Маянк Шарма (Mayank Sharma), Шашанк Шарма (Shashank Sharma), Ник Вейч (Nick Veitch), Коэн Верлоесем (Koen Vervloessem), Евгений Балдин, Артем Зорин, Михаил Пожидаев, Андрей Ушаков, Алексей Федорчук

Художественный ассистент Кейт Мак-Доннелл (Kate McDonnell)

Иллюстрации Шейн Коллиндж (Shane Collinge), Еллы Уолтон Иллюстрации (iStockPhoto), Саймон Миддлвек (Simon Middleweek)

Фото Джейсон Каплан (Jason E. Kaplan)

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 МЭИ)

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 эксклюзивное 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 ответственность 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 журналах, издаваемых 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

16+



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 ИСПОЛЗУЮТ:
 МВД РФ, Минздравсоцразвития РФ,
 Минфин республики 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



СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ВИРТУАЛИЗАЦИЯ:
**ЭКОНОМИЯ НА СТОИМОСТИ
IT-ИНФРАСТРУКТУРЫ ДО 90%**



Единая инфраструктура
на базе свободного
программного обеспечения

Нет лицензионных платежей —
расходы только на внедрение
и техническую поддержку

Минимальные затраты
на оборудование
за счет виртуализации



[www.linuxcenter.ru/shop/
linux-software/office/kitezh](http://www.linuxcenter.ru/shop/linux-software/office/kitezh)

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center