10 страниц развлекухи с Raspberry Pi

LINUX FORMAT

Главное в мире Linux

Апрель 2013 № 4 (169)

НА ВАШЕМ БЕСПЛАТНОМ DVD
Наша Fedora 18: с Gnome, KDE и Xfce
Плюс: CrunchBang, UberStudent и более того!

Steam для Linux!

» Готовьтесь к очередной игровой революции с. 46

Linux — что это?

Откройте, как работает лучшая в мире операционная система

ПЛЮС! Ускорение графики: от Homo Ludens – к Homo Sapiens с. 86

Репозитории Кладезь программ
X.org О языке родных осин
Django Советы на ход ноги

Fedora 18: мин нет?

» Берите самую новую программу с бета-тестирования Red Hat с. 14

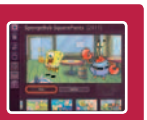
Бархатный диктатор

«Я вижу бурный рост web-мира Python, а не только Django»

Джейкоб Каплан-Мосс: прогноз для Python с. 38

Также в номере...

Ubuntu на Nexus
Превратите свой планшет с Android в полноценную Linux-машину с. 42



Сделаем ТВ-рекордер
Записывайте и вещайте HD-ТВ-программы с Raspberry Pi с. 64

Раскрутка личного облака
Это *ваши* данные — зачем хранить их у дяди? с. 78

Восстановление данных
SOS системы
» Обезопасимся от катастроф с данными

Мультимедиа
Музыка команд
» Рулите музыкой без менеджера окон

Системное администрирование
Как работают сети
» Берем две консервных банки, моток веревки и свистульку...

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

Linux Center
www.linuxcenter.ru

4 607130 825167 13004

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

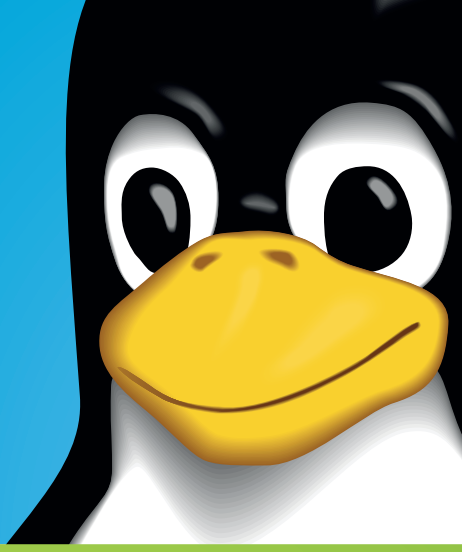
Москва +7 (499) 271-49-54

Санкт-Петербург +7 (812) 309-06-86

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

Linux center

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

Забыть 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 дружественных 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

1С:ПРЕДПРИЯТИЕ 8

Новое поколение систем автоматизации управления и учета

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

Теперь работает в Linux!

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

Сделайте шаг для работы вашей компании на Linux!

1С:ФРАНЧАЙЗИНГ

Mandriva

Etersoft

Москва +7 (499) 271-49-54

Санкт-Петербург +7 (812) 309-06-86

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

Linux center

LINUX FORMAT
Главное в мире Linux

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

Глубинное исследование нашей любимой ОС заставило нас оценить ее даже выше. Вот мы и спрашиваем: «Что в Linux нравится вам больше всего»?

Гэри Уокер
Lincity. Новый выпуск едва не стал хитом продаж: затягивает дьявольски!

Эндрю Грегори
Возможность менять тему рабочего стола по цветам успешных футбольных команд. Голубой от «Челси» – это ж прошлый сезон.

Эфраин Эрнандес-Мендоса
Все, кроме нехватки присутствия Шакиры. А то – Агилера там, Ханна Монтана тут...

Бен Эверард
Бородачи.

Маянк Шарма
По злой иронии судьбы, мои инициалы – MS, но я все равно очень люблю акронимы.

Джонатан Робертс
Майк Сондерс.

Майк Сондерс
Изобилие опций эмуляции для ретро-стальгических эпопей *Frontier: Elite II*.

Валентин Синицын
То, что теперь в нем, наконец, можно поиграть в *Half-Life* и *Counter-Strike*.

Ник Вейч
Linux бесплатен, и мне остается больше денег на поездки по свету вслед за гастролями Кристины Агилеры.

Сюзан Линтон
Лучшее в Linux – всякие дурацкие имена. *GIMP*? *QfFPSGUI*? Я их собираю.

Шашанк Шарма
Элегантный дизайн, чувство превосходства, гениальная панель, всемирная марка... э, а в каком я журнале?

Нейл Ботвик
Linux Format! Ну, а теперь мне прибавят, наконец, зарплату?

Linux – это...

» Вопрос, вынесенный на обложку этого номера, не имеет однозначного ответа. Все зависит от точки зрения. Для пользователя или сисадмина, Linux прежде всего репозиторий (на DVD или в Сети) и набор утилит установки и настройки. Для создателей дистрибутивов – постоянно обновляемый репозиторий и набор средств сборки и тестирования. Для разработчиков приложений – прежде всего удобная платформа для запуска средств разработки и самих приложений, причем в понятие «платформа», кроме ядра системы, входят библиотеки, фреймворки, графические среды и другие полезные компоненты. И, наконец, для авторов Linux – это проект создания ядра ОС общего назначения, причем один из самых успешных в истории ИТ-индустрии.

Обычно ответ на вопрос «Что это?» применительно к любой технической системе распадается на три части: зачем она нужна, из каких подсистем состоит и в какие надсистемы входит. Далее обычно идет комментарий о том, когда, кем и как она создана – речь тут уже не столько о технологии, сколько о людях и их организации в процессе производства.

А по-моему, Linux (в широком смысле) – это универсальный набор инструментов для решения любых задач современного ИТ. Причем для конкретной задачи всегда найдется их оптимальное сочетание и конфигурация. Создан этот набор благодаря уникальной самоорганизующейся системе взаимодействующих сообществ отдельных проектов. Именно она позволяет почти мгновенно реагировать на изменение потребностей пользователей и гасить пагубное влияние частных ошибок управления. Корпоративным бюрократиям такое и не снилось.

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

LINUX FORMAT

Содержание

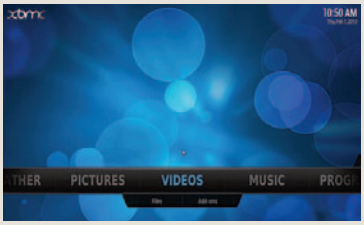
Сварено в Брюсселе, розлив Батский.

Обзоры

Fedora 18 14
Итак, релиз Fedora наконец состоялся. Посмотрим, стоило ли его дожидаться.

› **Под микроскопом — новый релиз Fedora, Spherical Cow [Сферическая Корова].**

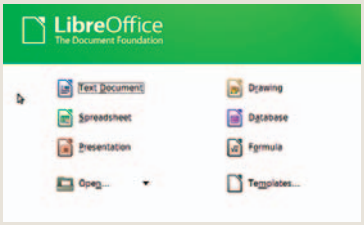
XBMC 16
Обзор этого нового медиа-проигрывателя счастливо обошелся без энтских жен.



› ***XMBC*** **— замечательный проигрыватель, и на стр. 16 вы узнаете, почему.**

CuBox 17
Читайте о наших приключениях с самым свежим мини-компьютером с Linux.

Libre Office 18
Мы оценили киловатты улучшенных функций интероперабельности *LibreOffice*.



› ***LibreOffice*****: свезло! Добавилась венгерская кодировка!**

Legend of Grimrock 19
Горячий финский вариант ролевой игры типа наведи-и-щелкни.

Pocketbook Touch 20
Электронные чернила – это сила! Они не выцветают на солнце.

Linux — что это?

Присоединяйтесь к нашей экскурсии в самое сердце лучшей в мире ОС

с. 30

Сравнение: Дистрибутивы для RPi с. 24

Что за штука — WebRTC? с. 56

Люди говорят

> Думаю, одно из преимуществ Django — здесь шире почва для эксперимента

Джейкоб Каплан-Мосс — об успехах Django с. 38

На вашем бесплатном DVD

Fedora 18
» Вариация Сферической Коровы от LXF

CrunchBang
» Продвинутая простота

ПЛЮС: UberStudent и прочее с. 106

Ищите в этом номере

Ubuntu на Nexus 7 42
Превратите свой планшетник в Linux-машину.

Steam для Linux 46
Провозгласим новую игровую революцию.

Репозитории 50
Там, где бережно хранятся тысячи пакетов.

Аз, буки – веди 52
Кириллица пробралась в *X.org* не сразу...

Пропустили номер?

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

Учебники

Raspberry Pi
Медиа 64
Превратим мини-компьютер в мощное средство видеозаписи ТВ-программ.

Сети
Создание сети 68
Как сделать маршрутизатор из вашего компьютера с Linux.

Востановление данных
SystemRescueCD 72
Live-дистрибутив, который поможет при кризисе системы и катастрофе с данными.

Командная строка
Beets 74
Способ отличить файлы MP3 «Сахарный пакет» от «Дискотека Авария».

Облачное хранение
ownCloud 78
Строим свободную альтернативу популярных онлайн-сервисов.

Web-разработка
Django 82
Закругляем серию полезными советами профессионала.

Вычисления
CUDA 86
Не играми едиными живы ускорители графики – хотя, конечно, все начиналось с игр.

Виртуализация
Офис на Virtual Box 90
Из одного Ubuntu получилось шесть Windows XP. А лицензии – в макулатуру!

Дистрострой
Сам себе дистрибутив 94
Вырастим собственное деревце – создадим дистрибутив с нуля.

Постоянные рубрики

Новости 6
Брюссель принял FOSDEM – Москва примет ROSS, Google оплачивает RPi, Canonical открыл новый *Mir*, Hurd скорее жив, а Samsung продожает судиться с Apple.

Новости Android 22
Телефон летает в небесах, а у Google проблемы с Поднебесной.

Сравнение 24
Подберите лучшую операционную систему для вашего Raspberry Pi. Android, Arch, Plan 9, Raspbian, Risc OS...

Интервью LXF 38
Образование филолога позволяет **Джейкобу Каплану-Моссу** писать превосходную документацию.

Что за штука 56
WebRTC намеревается обеспечить коммуникации в реальном масштабе времени и содружество браузеров.

Рубрика сисадмина 58
Предпоследняя часть серии про LAMP, на букву M: Доктор берется за *MySQL*.

Ответы 96
ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!
Нейл Ботвик – про KDE и Mint, настройку Unity, пароль для защиты *Grub*, имена хостов Android и прочее.

Hotpicks 100
Отведайте горяченького: лучшие в мире новинки свободного ПО. *FatRat*, *Avogadro*, *Safe-rm*, *XOWA*, *Cppcheck*, *Parcellite*, *PokerTH*, *Erebus*, *Gtkdialog*, *LiteIDE*, *Mednafen*.

Диск *Linux Format* 106
Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108
Слава прежних выпусков не померкла: они как попало свалены на ничейной территории, некогда бывшей столом Джона.

Через месяц 112
Взлом Сети, *Samba 4*, *XBMC* и дистрибутивы для продвинутых пользователей.

ГЛАВНОЕ FOSDEM'13 » Детям, детям » Новый Мир » Огнелис в телефоне » Свобода не за горами » WebOS и LG » Патентные войны

ФОРУМ РАЗРАБОТЧИКОВ

FOSDEM'13: 486 докладов!

» Рубрику готовил АРТЕМ ЗОРИН

Столица Бельгии на 2 дня стала столицей технарей.

2 и 3 февраля 2013 года свыше 5000 разработчиков ПО с открытым исходным кодом со всей Европы – и не только – съехались на FOSDEM 2013, чтобы слушать доклады и участвовать в дискуссиях о разработке свободного/открытого ПО и обсудить развитие отрасли. В основном это были «технари». Мероприятие состоялось в кампусе Свободного университета Брюсселя [Université Libre de Bruxelles].

В этом году можно отметить три ключевых доклада: «Как мы построили сообщество Jenkins», представленный Косуке Кавагучи [Kohsuke Kawaguchi], лидером проекта Jenkins (ветки системы непрерывной интеграции Hudson); «Дьявол кроется в деталях» – о том, что мешает развитию и распространению свободного/открытого ПО, сделанный Амелией Андерсдоттер [Amelia Andersdotter], членом Европарламента, вовлеченным в продвижение СПО в Евросоюзе; и доклад под названием «Хранитель секретов» от Лесли Хауторн [Leslie Hawthorn] – сотрудника Red Hat, отвечающего за взаимодействие Red Hat и сообщества, посвященный трудностям взаимодействия и межличностного общения между различными членами проектов с открытым исходным кодом; в частности, она уделила особое внимание проблемам прозрачности и открытости в сообществах СПО.

А всего на конференции прозвучало 486 докладов от 474 разных спикеров, разделенных на 40 основных тем и группированных по алфавиту – от языка программирования Ada до графического сервера *X.org*. Среди них были, например, прекрасный доклад «Простые способы взлома для хакеров-новичков» Майкла Микса [Michael Meeks], одного из лидеров проекта *LibreOffice*, и углубленная техническая дискуссия по архитектуре Neo4j HA – высокопроизводительной, не-SQL базы данных, основанной на принципе графов. Одной из наиболее страстных и оживленных была речь Кэти Мэлмроуз [Cathy Malmrose] (генерального директора компании ZaReason, выпускающей оборудование, адаптированное под открытое/свободное ПО), посвященная приснопамятной технологии UEFI Secure Boot.

Из названия мероприятия следует, что основными темами разговоров являются сугубо технические вопросы, но были доклады и на общие темы, так что форум не выглядел таким уж «технарским». Трудно было найти тему, интересующую сообщество открытого/свободного ПО, которая не была бы освещена на конференции. Каждый мог найти для себя что-то интересное, чтобы с пользой провести эти два дня. Вход на мероприятие был бесплатным (как и пиво и кофе в произвольном количестве!), поэтому участники толпами бродили между «комнатами» без всяких ограничений.

Большинство посетителей FOSDEM'13 осели в т. н. «программерских комнатах», где ключевые фигуры открытых проектов вели переговоры и дискуссии.

> «Трудно найти тему, которая не была бы освещена на конференции.»

Благодаря прекрасной слаженной работе команды по обработке видео, многие из обсуждений были оперативно записаны и закачаны на сайт video.fosdem.org. Там же можно посмотреть и скачать видео с FOSDEM предыдущих лет.

Помимо обсуждений и докладов, многие крупные сообщества СПО представили свои тематические стенды, на которых посетители могли пообщаться с разработчиками, задать им интересующие вопросы.

› Mozilla начинает дискуссию-доклад, представляя свои телефоны на Firefox OS.

Наибольший интерес публики вызвал стенд сообщества Mozilla, выставившего несколько телефонов под управлением Firefox OS на пробу всем желающим. Притягательным был стенд Pandora – компании, выпускающей мобильный игровой компьютер с контроллерами для игр под управлением Linux; его тоже можно было потрогать и поиграть на нем.

Ключевой особенностью любой конференции разработчиков СПО являются не доклады, даже если они очень интересны, но шанс живого общения и взаимодействия с другими членами сообщества. На FOSDEM'13 это происходило в пределах стендов, в «комнатах» разработчиков, и, после официальной части, в многочисленных барах Брюсселя. Куда ни глянь – всюду «технари», обсуждающие тонкие и не очень вопросы разработки и развития СПО. Кофе и пива хватало для поддержки дискуссий на достойном уровне.

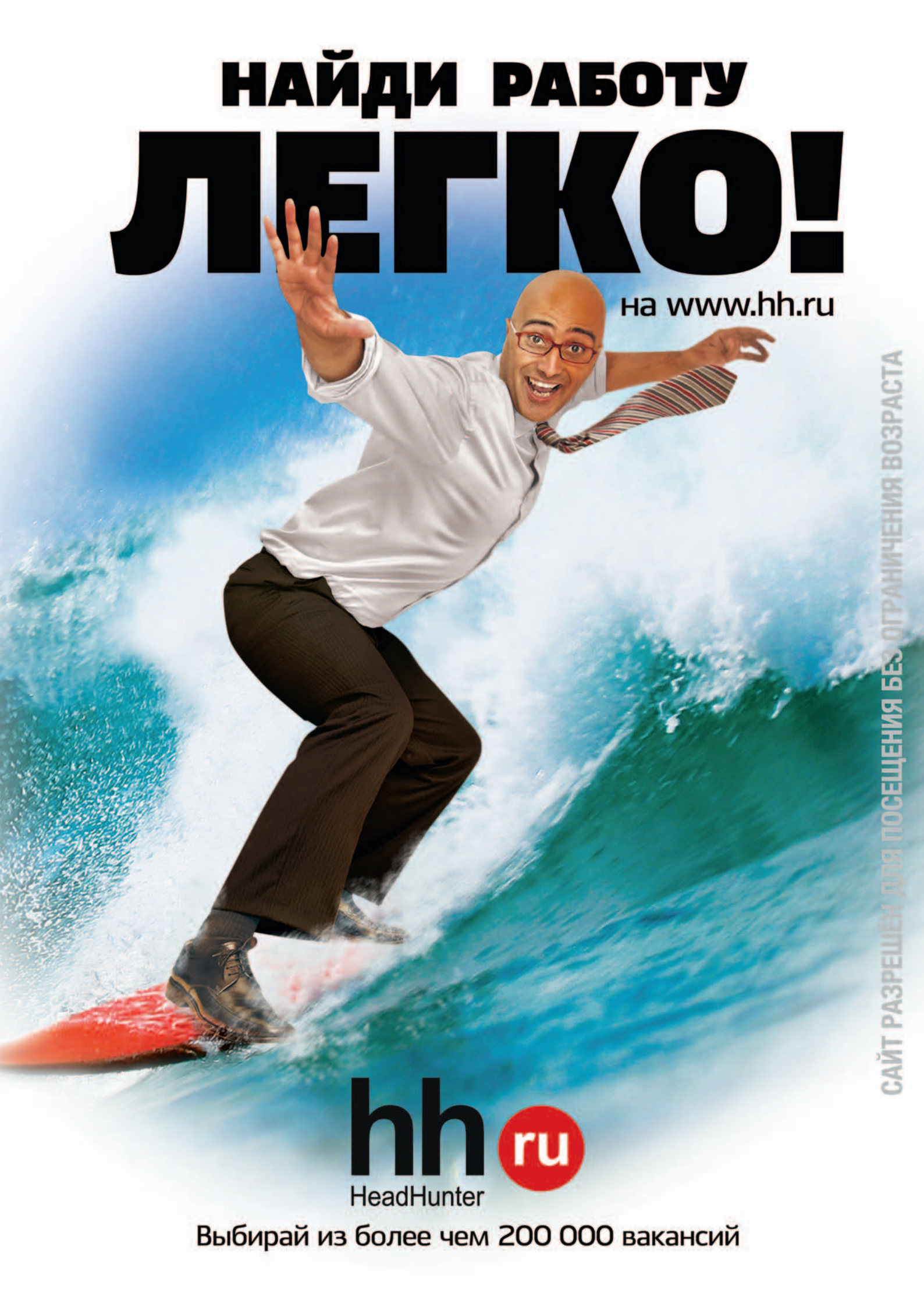
На FOSDEM'13 побывал и наш соотечественник – Дмитрий Мережковский, участник проекта Aldebaran Robotics по разработке человекоподобных роботов.

По словам Бена Эверарда [Ben Everard], обозревателя портала tuxradar.com, двух дней, отведенных на FOSDEM'13, явно не хватило, чтобы посетить даже самые интересные события конференции. С нетерпением ждем следующего года.

Фото © http://a.yfrog.com

НАЙДИ РАБОТУ ЛЕГКО!

на www.hh.ru



САЙТ РАЗРЕШЕН ДЛЯ ПОСЕЩЕНИЯ БЕЗ ОГРАНИЧЕНИЯ ВОЗРАСТА

hh.ru
HeadHunter

Выбирай из более чем 200 000 вакансий

КОМПЬЮТЕР ДЛЯ ДЕТЕЙ

Google платит за RPi

15 000 компьютеров-малюток скоро пойдут в школу.

Тысячи детей в Англии скоро получат бесплатные компьютеры Raspberry Pi, благодаря успешному сотрудничеству между Google и Raspberry Pi Foundation. Первыми счастливчиками стали члены кембриджского колледжа Честертон [Chesterton Community College].

Выбор Google пал не Англию не случайно: ведь именно здесь Raspberry Pi был разработан как образовательный инструмент, но в основном для университетов, а не для колледжей и школ. Исполнительный директор Google Эрик Шмидт [Eric Schmidt] решил восстановить добрую традицию Google по финансированию образовательных учреждений, подзабытую из-за проблем Google с британскими налоговыми органами. «Мы работаем над появлением нового поколения ИТ-специалистов, и надеемся, что поставка Raspberry Pi в британские школы и колледжи поможет создать новую волну инноваций», сказал Шмидт, вспомнив славную историю английских изобретателей.

RPi имел в Британии огромный успех: продано уже более миллиона экземпляров. Основатель RPi Foundation Эбен Аптон [Eben Upton] говорит: «Надеемся, что наше сотрудничество с Google станет важным моментом в компьютерном образовании страны. Мы считаем, что это поможет развернуть печальную тенденцию падения навыков у студентов, и надеемся, что у нас появятся новые люди, изменяющие мир к лучшему – ведь телефон, ТВ и компьютеры были изобретены здесь, в Британии».

Директор образовательного департамента Microsoft Стив Бесвик [Steve Beswick], увлеченный идеей выдачи образовательных кредитов, считает компьютерные науки для Microsoft столь же важными, как физика, химия или биология: «Мы долго считали ИТ-науки прикладными, что оказалось неверным. Если учить детей информатике с первого класса, выше вероятность увеличения числа тех, кто будет изучать предмет в университете и в конце концов работать в этой сфере».

› 15 000 RPi покрывают малую толику неуплаченных налогов огромной компании...

Представитель Microsoft отчасти лукавит: не в интересах Microsoft взращивать программистов, склонных программировать на открытых платформах. Но это и не в интересах Google. Скорее – это в интересах развития инноваций во всем мире.

Заметим, что стоимость всех 15 000 RPi, поставку которых в британские школы оплатила Google, составляет 350 000 фунтов стерлингов, и это всего лишь 0,175 % от общей суммы налогов, в неуплате которых на территории Соединенного Королевства обвиняется компания.

ИННОВАЦИОННЫЙ CANONICAL

Ubuntu продолжает удивлять

Анонсирован дисплейный сервер *Mir*, альтернатива *Wayland* и *X.org*.

Canonical представила план создания графического на базе нового дисплейного сервера *Mir*, который будет применен в Ubuntu вместо *X.org* и *Wayland*. *Qt* станет основным инструментом для разработки приложений и графической оболочки дистрибутива.

Одновременно появился проект Unity Next по кардинальной внутренней переработке оболочки Unity. Главное его нововведение – переход с *GTK+* и *OpenGL*-инструментария *Nux* на инструменты *Qt5* и средства декларативного описания интерфейсов QML. Тогда Unity получит единую кодовую базу для формирования окружений в различных классах устройств, автоматически адаптируясь к широкому диапазону разрешений экрана (от смартфонов до ПК), сохранив прежний внешний вид и особенности использования.

Mir состоит из двух базовых библиотек: *libmir-server* с набором функций для реализации композитных серверов и *libmir-client* с реализацией клиентского интерфейса, позволяющего приложениям взаимодействовать с сервером. На базе *libmir-server* развивается композитный сервер *unity-system-compositor*.

До сих пор *Mir* развивался сугубо внутри Canonical. Пакеты с начальной реализацией *Mir* теперь доступны для загрузки с сервиса Launchpad, но лишь базовый каркас: большая часть функциональности еще только в планах. Код *Mir* распространяется под лицензией LGPLv3 и насчитывает около 15 тысяч строк на языке C++.

Интеграцию *Mir* в Ubuntu и наращивание функциональности нового графического стека планируется производить поэтапно. Первый прототип Unity Next, основанного на *Mir*, будет доступен для тестирования в начале мая 2013 года. В октябре планируется обеспечить полную интеграцию *Mir* и Unity Next с дистрибутивом и ввести их в редакции Ubuntu для смартфонов. В апреле 2014 года новое окружение будет доступно для всех платформ, включая ПК, планшеты, телевизоры.

По словам Canonical, в большинстве случаев использование *Mir* внешне не отразится на взаимодействии пользователя с системой. Оболочка Unity с виду останется неизменной. Будет продолжена поддержка запуска приложений *X11*.

Работа *Mir* организуется поверх существующей инфраструктуры видеодрайверов. Будут поддерживаться как свободные, так и проприетарные драйверы, а также драйверы платформы Android.

Canonical ранее рассчитывала перейти на *Wayland*, но разработчиков Ubuntu не устроила организация обработки событий ввода, во многом повторяющая подход *X.Org* и затрудняющая построение современных систем для мобильных устройств с сенсорными экранами.

У некоторых разработчиков *Wayland* и *X.org* идея создания *Mir* уже вызвала негативную реакцию.

Фото © http://documentally.com

softline®

Services Software Cloud

ИТ-архитектура вашего бизнеса

CLOUD

SaaS

SOLUTIONS

remote workplace

PROJECTS

VIRTUALIZATION

hardware

legal support

education

Mirapolis

SECURITY

учебный центр

ERP&CRM

CAD
CAM
CAE

защита персональных данных

SOFTWARE

техническая поддержка

data leak prevention

корпоративный портал DeskWork

open source

SAM

юридическая поддержка

LINUX

программное обеспечение

SERVICES

САПР и ГИС

информационная безопасность

search

flexible

CONSULTING

решения Microsoft

PLATFORM

protection

hosting

technical support

+7 (495) 232-00-23 www.softline.ru info@softline.ru

ТЕЛЕФОН ДЛЯ БИЗНЕСА

Компании подтягиваются

Sony объявила о разработке телефона на базе Firefox OS.

В ходе прошедшего в Барселоне 25–28 февраля конгресса мобильных технологий (MWC 2013) компания Sony объявила о намерении сотрудничать с испанским оператором связи Telefónica и выпустить на рынок телефон на базе открытой мобильной платформы Firefox OS. Говорится, что новое устройство поступит на прилавки в 2014 году. В качестве мотива выпуска телефона с Firefox OS называется высокий потенциал платформы и мобильных технологий на базе HTML5, впервые в индустрии построенных полностью на базе открытых стандартов и развиваемых независимо от отдельных производителей.

Одновременно компания Sony выпустила тестовый образ прошивки на базе Firefox OS для смартфона Xperia E и опубликовала инструкции по разблокировке телефона и установке тестовой прошивки, а также краткое введение по разработке приложений для Firefox OS. Сборка подготовлена в процессе внутренних экспериментов инженеров Sony с Firefox OS. Прошивка позиционируется для начального ознакомления и не предназначена для повседневного использования. Например, из-за отсутствия сертификации не активирована работа телефонии, Wi-Fi и Bluetooth, имеются проблемы с калибровкой экрана и могут наблюдаться проблемы со стабильностью при работе с SD-картой.

Это был третий анонс смартфонов на базе Firefox OS, прозвучавший на MWC 2013. В первый день выставки компании Alcatel и ZTE представили телефоны Alcatel One Touch Fire и ZTE Open. В качестве одного из важных конкурентных преимуществ упоминается низкая цена, делающая устройства доступными широкому кругу потребителей.

Кроме того, на выставке было объявлено скорое начало продаж двух моделей телефонов, подготовленных проектом Geeksphone совместно с компаний Telefónica и ориентированных на разработчиков приложений для Firefox OS. Модель Keon позиционируется как простой и недорогой телефон. Модель Peak имеет более продвинутые характеристики. Стоимость устройств пока официально не объявлена, но по словам представителя компании

› Смартфон Sony может стать флагманом среди телефонов на Firefox OS.

модель Peak будет стоить в районе €200, а модель Keon – чуть больше €100.

Телефон Xperia E позиционируется Sony как бизнес-смартфон среднего уровня.

Ранее 18 производителей оборудования и операторов связи объявили о своей заинтересованности в мобильной платформе Firefox OS, среди которых особо стоит отметить российские Мегафон и Билайн.

ВОЗВРАТ К КОРНЯМ

2013 – год Hurd на десктопе?

Полностью свободная ОС из далеких 1990-х...

На прошедшем FOSDEM'13 разработчики полностью свободной ОС GNU/Hurd объявили, что следующим этапом в развитии проекта станет поддержка USB-портов и жестких дисков SATA. Основным докладчиком выступал Самюэль Тибо [Samuel Thibault], активный разработчик Debian и GNU/Hurd; он представил свой взгляд на будущее Hurd и ознакомил посетителей с некоторыми ключевыми целями проекта на 2013 год.

Легче всего будет попробовать GNU/Hurd в предстоящем релизе Debian Wheezy, в котором будет реализована неофициальная поддержка запуска свободного ядра. По словам Тибо, *Xfce* уже работает с Hurd, в то время как работы по портированию окружений Gnome 3 и KDE близятся к завершению. Планы на будущее включают поддержку архитектуры x86_64 и официальную поддержку и включение GNU/Hurd в следующий после Wheezy релиз Debian 8.0 под кодовым именем Jessie.

Прогресс в разработке GNU/Hurd был достигнут в начале года, когда кураторы проекта объявили о работе по портированию компонентов LLVM, в том числе компилятора *Clang*, для работы в окружениях на базе GNU/Hurd. Работа по портированию ведется в тесном сотрудничестве с разработчиками проекта LLVM. Пока портирование завершено не полностью, но порт уже можно считать более-менее работоспособным, за исключением некоторых нюансов. Это даст возможность портировать многие приложения Linux на GNU/Hurd. Уже сейчас готовы почти все компоненты Hurd для полноценной работы в связке с Debian: работающий инсталлятор с возможностью установки в графическом режиме, а также по сети; почти 80 % пакетов Debian теперь работают в Hurd; обеспечена поддержка почти всех основных классов устройств.

Главное отличие (а для многих – преимущество) GNU/Hurd – это ее микроядерная архитектура. Основа Hurd – микроядро GNU Mach, которое успешно используется Apple Mac OS X с 2000 года.

Разработка GNU/Hurd ведется с 1990 года. По замыслу создателей, Hurd должен был стать ядром полностью свободной ОС GNU. Однако выход Linux в 1991 году сильно затормозил развитие Hurd, так как благодаря Linux отпала необходимость в написании ядра с нуля.

Фото © www.sibir-ix.ru

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

redhat
PREMIER BUSINESS PARTNER

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

ИНОГДА ОНИ ОЖИВАЮТ

Плаформе — жить

LG выкупила webOS у HP и хочет использовать ее в своих телевизорах.

Компания Hewlett-Packard продала LG Electronics активы, связанных с разработкой открытой мобильной платформы webOS. LG получила права на код, документацию, торговые марки, сайты проекта и группу инженеров-разработчиков, а также лицензию на использование патентов, приобретенных HP вместе с компанией Palm. Финансовые детали сделки не разглашаются. HP сосредоточится на стратегических направлениях, связанных с облачными технологиями.

LG планирует развивать платформу webOS для создания, в частности, умных телевизоров нового поколения. Инвестиции в развитие данной технологии помогут компании добиться лидерства в сфере потребительской электроники с поддержкой интегрированных средств для работы с web-сервисами. LG намерена стать лидером проектов Open WebOS и Enyo, сохранив их открытый статус.

Платформа webOS использует для разработки приложений и базового интерфейса технологии CSS, HTML5 и JavaScript

Многострадальная webOS просто обязана прижиться на умных телевизорах.

и JavaScript-фреймворк Enyo, который предоставляет набор виджетов и макетов размещения элементов [layout].

Основа webOS – системный менеджер *Luna* на базе *Qt* и *QtWebKit* – управляет выполнением web-приложений, меню и статусной строкой, организует работу интерфейса Card View, обеспечивает запуск приложений, формирует экран блокировки системы. *Luna* обеспечивает запуск пользовательских и системных приложений (почтовый клиент, календарь-планировщик, система ведения заметок, адресная книга, часы, калькулятор, менеджер финансов и web-браузер Isis).

ДЕЛА СУДЕБНЫЕ

Троллей осадили

Суд решил не менять размер выплаты Samsung в пользу Apple.

Ходатайство Apple об увеличении выплаты Samsung до $1,05 миллиарда отклонила та же судья Люси Ко [Lucy Koh], что назначила этот беспрецедентный в истории патентных войн штраф. Ко объяснила: «Учитывая, что Apple так и не смогла ясно указать, какой именно ущерб нанесло якобы копирование товарного вида телефонов Apple компанией Samsung, суд оставил размер компенсации на свое усмотрение, и не видит необходимости в ее повышении». Суд выяснил, что Apple не представила убедительных доказательств «объективно высокой вероятности нарушения компанией Samsung патентов Apple». Но ряд мобильных устройств Samsung запрещены к продаже в США.

Вердикт в пользу компании Apple, выдвинувшей иск против Samsung с тем, чтобы запретить продажи смартфона Samsung Galaxy Nexus и других устройств на Android на всей территории США, угрожал экосистеме Android в целом. Но суд г. Сан-Хосе (Калифорния) отменил это абсурдное решение.

Телефоны Nexus для Google стала выпускать и LG – видимо, Google «подстраховался» на случай, если Apple снова задумает блокировать продажи.

По оценке Колин Чэнь [Colleen Chien], профессора Университета Санта-Клары (Калифорния), большинство патентных разбирательств в США инициируется патентными троллями, которые ничего не производят и не изобретают. Пять лет назад таких дел было 23 % , в 2011 г. – 45 %, а в 2012 – 61 %. Лишь одно из нескольких сот доходит до суда, прочие решаются соглашением о выплатах. LXF

Новости короткой строкой

12.04.2013 в отеле «Ренессанс» (Москва) пройдет Russian Open Source Summit 2013. Ключевые вопросы: место и роль ИТ в реализации стратегии кибербезопасности государства, бизнеса и общества; сертификация СПО-продуктов. Регистрация: http://www.pcweek.ru/foss/conference/

Началось открытое бета-тестирование Linux-версии плагина ОРФО-Lite для офисных пакетов *LibreOffice* и *Apache OpenOffice*, с модулем проверки орфографии русского языка согласно рекомендациям комиссии РАН.
Источник: www.informatic.ru/libre-beta

Компания Adobe выпустила *Photoshop Touch* для смартфонов на Android 4.x, стоимостью $4,99.
Источник: http://blogs.adobe.com

Еврокомиссия оштрафовала Microsoft на $732 млн за нарушение подписанного в 2009 году соглашения о предоставлении выбора web-браузера при установке ОС Windows.
Источник: http://europa.eu

Apache OpenOffice 3.4 преодолел отметку в 40 миллионов загрузок, с момента запуска в мае 2012 года.
Источник: www.openoffice.org

Компания Opera Software объявила о начале тестирования нового мобильного браузера, использующего движок *WebKit*. Браузер пока доступен только в версии для платформы Android.
Источник: http://business.opera.com/press/releases

Valve сообщает, что 2,02 % всех активных пользователей Steam используют Linux. По сравнению с январем доля линуксоидов удвоилась.
Источник: http://store.steampowered.com

Mozilla Foundation не собирается выпускать версию *Firefox* для Apple iOS, пока Apple не смягчит свою позицию по отношению к сторонним браузерам.
Источник: www.mobiset.ru

Из Google Play удалены все приложения для блокирования показа рекламы, в т. ч. популярное AdBlock Plus.
Источник: https://adblockplus.org/blog

Google и MPEG LA заключили лицензионное соглашение, и MPEG LA прекращает формирование патентного пула для борьбы с VP8. Кодек WebM теперь лицензионно безопасен.
Источник: http://blog.webmproject.org

Фото © http://stg.lgsolutions.com. В данном выпуске новостей использованы материалы с сайтов: www.opennet.ru, www.km.ru, www.linux.org.ru, www.mobi.ru, www.ruwikipedia.org, www.wsj.com, www.linuxformat.com, www.tuxradar.com, www.slashdot.org и др.

LINUX FORMAT

Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов

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

Файловая система для SSD

За последние лет 10–15 мы не раз читали победные реляции об успехах в компьютерной сфере. Однако настоящим успехом можно считать только начало широкого распространения SSD-накопителей. Впервые за всю историю дисковая подсистема перестала хронически тормозить производительность, благодаря трем факторам: ценовому, хардверному и софтверному.

С первым все ясно: нынче не нужно быть Крезом, чтобы позволить себе в ноуте или десктопе такой накопитель на 120–240 ГБ, чего достаточно для системы, приложений и текущих рабочих данных (при условии, что система эта – Linux). С хардверным – тоже: самый бюджетный SSD нынче оставляет традиционные винчестеры в... далеко. Главное же – рост надежности: первые SSD вызывали обоснованные опасения за их долголетие, но сейчас в десктопном сегменте эта проблема практически снята. А софтверный заиграл только сейчас. Конечно, в традиционных файловых системах поддержка твердотельной специфики появилась достаточно давно, но с рядом ограничений. А ФС, изначально разрабатывавшиеся для SSD (вроде LogFS), распространения не получили.

И вот – свершилось: в ядро Linux версии 3.8 штатно включена поддержка F2FS – «файловой системы, дружественной к флэшкам». Разумеется, она еще ждет «обкатки» во всех отношениях. Но то, что разработана она фирмой Samsung, одним из основных производителей SSD, позволяет надеяться на то, что обкатка эта долгой не будет.

alv@posix.ru

Сегодня мы рассматриваем:

Fedora 18 14
Вообще-то мы любим Fedora: этот полигон передового ПО дает нам почувствовать себя исследователями или учеными, отважно раздвигающими границы человеческих знаний. Но не этот релиз. На сей раз исследователи получают лживый компас и сомнительную карту.

XBMC 12 16
Раз уж закрылся наш местный пункт проката DVD, нам понадобился способ направлять поток живого ТВ на экран, записывать его, добавлять меню и управлять зомбоящиком со всеми удобствами смартфона на Android. И – ура! Во исполнение пророчества явился *XBMC*, способный показывать драмы сезона даже в их неурочное время.

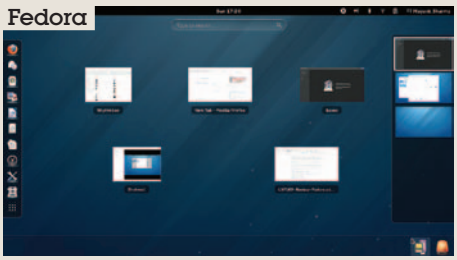
CuBox 17
Хорошенького должно быть понемножку. Знания высшего порядка прибывают в форме мистического черного монолита. Скрестите эти два высказывания в материальном варианте – и получится славный CuBox, крошечный компьютер на базе ARM.

LibreOffice 18
Нудный, нудный *LibreOffice* потихоньку отвоевал позиции самого компетентного, надежного и полезного свободного программного комплекта. Подобно Святому Петру, это камень, на котором мы строим нашу веру. Данный релиз проявил заботу и об освобождении язычников...

Legend of Grimrock 19
В этом проекте весьма удачно сложились все необходимые элементы упоительного исследования подземелий. Каковому немало способствует теплая компания тщательно отобранных соратников – монстров, минотавров, магов... а до кучи воров и воинов. Итак, вперед, в подземелья! Или лучше сказать – «вниз»?

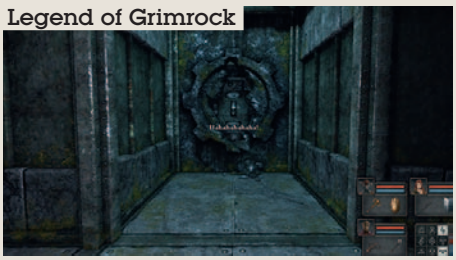
Pocketbook Touch 20
На фоне собратьев данная читалка электронных книг «цепляет» отличным экраном, легко читаемым и в полутемном вагоне метро, и на пляже в солнечный день. Кроме того, в ней предусмотрены словарь, web-браузер, игры, музыкальный плеер – и много чего еще!

Fedora



› **Fedora, мы критикуем тебя, потому что любим и хотим быть с тобой честными.**

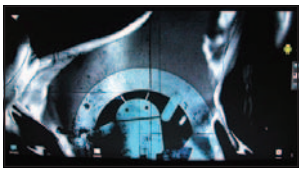
Legend of Grimrock



› **Ваши попытки прорыва сквозь подземные лабиринты встречает леденящий душу хохот Главгада.**

Сравнение: Операционки для Raspberry Pi с. 24

Android



Arch

Plan 9

Raspbian

Risc OS

Разнообразие придает жизни вкус – не встряхнуть ли вам свой Raspberry Pi новой ОС? Зачем?.. Да просто потому, что вам это не слабо! Предлагаем пять лучших экземпляров для даровитых самоделкиных.

Fedora 18

LINUX FORMAT на DVD

Спустя пару месяцев задержки наконец вышел новый дистрибутив Fedora 18. Стоила ли ожиданий «Сферическая Корова», размышляет **Маянк Шарма**.

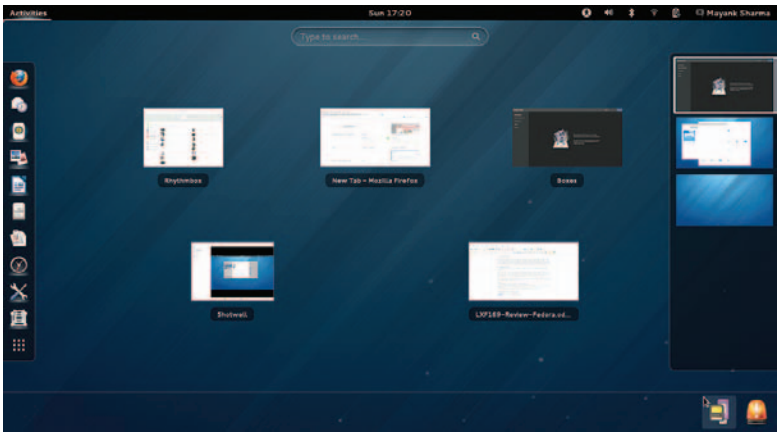
Вкратце

» Лидер дистрибутивов на базе Gnome, укомплектованный исключительно свободным ПО. Альтернативные RPM-дистрибутивы — OpenSUSE и Mageia.

Возможно, мы повторяем ведомое всем, но Fedora – это тестовая площадка для самых новейших и замечательных технологии, которые в итоге окажутся в Red Hat Enterprise Linux. Fedora – мощнейший дистрибутив именно для опытного линуксоида, т.к. по удобству работы уступает более понятным рядовому пользователю дистрибутивам Linux Mint и Ubuntu.

В Fedora 18 в этом смысле все по-прежнему. Отсутствует интеграция с «облачным» хранением данных, установка мультимедиа-кодеков требует танцев с бубном; не впечатлят рядового пользователя ни функции управления рабочим столом, ни причудливые программные сервисы.

Мы не рекомендовали бы ее новым пользователям Linux, но это не означает непригодности Fedora 18 для работы. Последняя версия предназначена для сосуществования с Windows 8 на новейших ПК с UEFI Secure Boot [безопасная загрузка].



› Изготовление приглядного экранного снимка рабочего стола Gnome в Fedora 18 заняло больше времени, чем написание всей статьи.

> «“Дистроперестройщики” могут подправить инсталлятор.»

В официальные репозитории введен инструмент *pesign*, благодаря которому системный администратор сможет издавать свои сертификаты и подписывать локальные изменения в *Grub* или ядре.

И хотя в этой версии при привязке к NTFS-разделам Windows 8 драйверы ntfs-3g работают лучше, разработчики советуют отключать новую функцию ‘fast restart’. При использовании двух систем, в момент перезапуска Windows 8 на Fedora 18 включенная функция ‘fast restart’ может привести к потере данных в разделах NTFS Windows 8.

Кусачая Anaconda

Одряхлевший инсталлятор *Anaconda* полностью переработан.

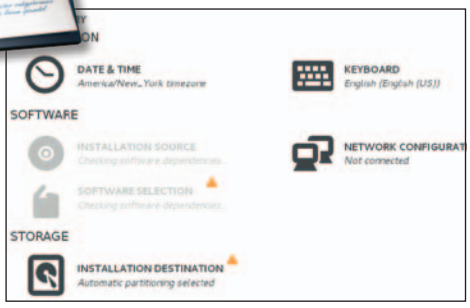
В предположении, что именно из-за этого задержали выпуск версии, естественно ожидать, что финальный продукт будет безупречен. Увы, случай явно не тот. Мы в Башнях **LXF** не охаиваем новую технологию только потому, что она нам непривычна. И заявляя, что мы разочарованы обновленным инсталлятором, мы говорим не о стиле, а по существу.

Разбиение диска на разделы на мультизагрузочных системах превратилось в сущий кошмар. Если дисков несколько, они будут идентифицироваться по дате создания и модели, а не по точкам монтирования. Если диски одинаковые, правильный выбор между ними сильно затруднен, как и случилось на одной из наших тестовых машин. Если потребуется изменить существующие разделы или освободить место, лучше использовать сторонний инструмент, например *Gparted*.

При запуске Fedora18 не удастся инсталлировать начальный загрузчик в корневой раздел. Если желаете установить *Grub*, то он встанет только на MBR диска. Опять же далеко не очевидна опция пропуска инсталляции начального загрузчика. При разбиении диска, другие дистрибутивы Linux (Ubuntu и Mint) и инсталляции Windows 7 на всех тестовых машинах были опознаны как неизвестные. Тем не менее они корректно отображались в стильном меню *Grub* Fedora 18.

Единственное преимущество нового инсталлятора в том, что он, в отличие от старого (где надо было методично отшагать по всем окнам), делает процесс установки более целеустремленным благодаря новой модели интерфейса пользователя. В Fedora 18 «дистроперестройщики» могут подправить инсталлятор массой способов. Также упрощен процесс создания логиче-

Свойства навскидку



Новый инсталлятор
Переработанный инсталлятор *Anaconda* стал болтливее, но менее понятный.

Больше рабочих столов
Помимо Gnome, есть еще рабочие столы Cinnamon, Mate, KDE и *Xfce*.

Fedora vs OpenSUSE 12.2

Как эти два гиганта делят между собой вселенную RPM? За спиной у обоих стоят корпорации, и оба поддерживают нескольких рабочих столов. Версия OpenSUSE 12.2 вышла тоже с опозданием, за пару месяцев до выпуска Fedora 18.

Отличительной чертой OpenSUSE всегда являлась настраиваемость, особенно благодаря *YaST*, инструменту установки и настройки. Хотя в Fedora есть несколько собственных специальных утилит конфигурирования, но нет единого приложения типа *YaST*, объединяющего их. Вдобавок Gnome System Settings подменили собой функциональность большей части системных инструментов Fedora, в то время как в версии OpenSUSE с Gnome предусмотрены указатели на *YaST*.

Не впечатлил и пользовательский интерфейс Fedora для *Yum*, менеджер пакетов *PackageKit*, вместо *Yum Extender*, который сейчас прилагается к Mate, рабочему столу Fedora 18. Включена также пробная версия *dnf*, ответвления *Yum*, позиционируемая как более эффективная и быстрая утилита управления пакетами.

В своем обзоре OpenSUSE 12.2 (**LXF164**) Грэм обратил внимание, что OpenSUSE делает заставки для окон приложений типа *GIMP*, чтобы рабочий стол смотрелся более гармонично. Fedora не уделяет особого внимания подобного рода художествам. В довершение всего, наши тесты показали, что Gnome, который исконно считался сильной стороной Fedora, лучше запускается на OpenSUSE, причем даже на виртуальной машине.

Короче, если вам требуется дистрибутив на базе RPM, рекомендуем OpenSUSE. Версия OpenSUSE 12.3 должна выйти к моменту прочтения вами данной статьи.

ских томов LVM, а группа томов автоматически регулируется при добавлении и удалении тома или изменении его размера.

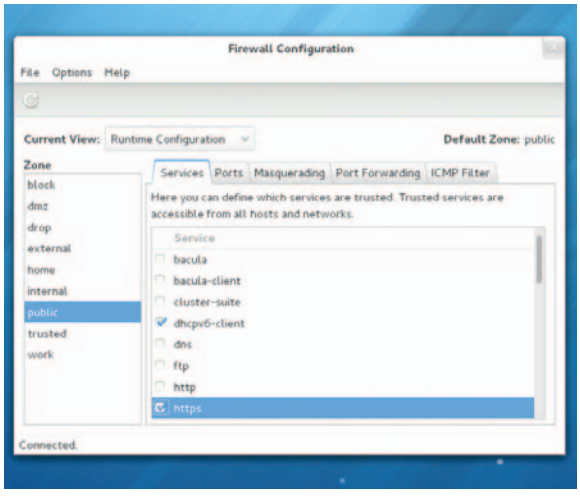
Действующим пользователям Fedora при обновлении до следующей версии рекомендуем использовать новый FedUP CLI. По умолчанию инструмент предназначен для извлечения пакетов с зеркал Fedora, но пригоден и для работы с образом DVD.

Еще в Fedora 18 представлен новый механизм обновления системы – Offline Updates. При готовности новых системных обновлений, в дополнение к уже представленной опции Power Off, Fedora 18 предложит опцию Install Updates&Restart. Это загрузит компьютер в среду с минимумом возможностей, позволив в момент обновления исключить конфликты с запущенными библиотеками и сервисами.

Радости Gnome

Большинством визуальных изменений Fedora 18 ожидаемо обязана рабочему столу по умолчанию – Gnome 3.6.

Появился новый инструмент *Boxes*, для удаленного доступа к другим машинам и локального создания виртуальных.



› В новом приложении *Firewall* нет опции отключения брандмауэра — мы считаем это довольно важным упущением.

Но опций настроек в нем очень мало, нет опций создания виртуальных дисков в произвольном месте и настройки сетевой конфигурации, нет возможности делиться файлами с хостом.

Приложение *Gnome Online Accounts* расширено, к тому же учетные записи Google теперь поддерживают Facebook, Microsoft Exchange и Windows Live. Еще есть *Nautilus*, отныне известный как приложение *Files*. Он щеголяет упорядоченным интерфейсом, удобным поиском файлов, новыми функциями запоминания последнего местоположения и активируемым пробелом просмотрщиком файлов *Sushi*.

Между тем возможности контекстного меню правой кнопки очень ограничены, нет опций создания файлов и документов. Но это все же лучше, чем ситуация с рабочим столом, который не реагирует на правый щелчок мыши и вообще не имеет контекстного меню.

Другим внешним изменением стал унаследованный из прошлой версии Gnome переработанный обзор деятельности Activities overview, экран блокировки, странно напоминающий аналог из Windows 8, и новая версия приложения *Clocks*.

Вне Gnome интеграция иных рабочих окружений невозможна. В Mate и Cinnamon по нескольку копий *GTK*-приложений, таких как менеджер архивов, калькулятор, терминал, словарь и другие. Некоторые инструменты Fedora 18, например SELinux (включаемый после установки пакета *policycoreutils-gui*), не запускаются из Mate, тогда как управление новым межсетевым экраном (*Firewall*) недоступно в KDE 4.9.

Корпоративная среда

Корпоративные пользователи – видимо, единственная категория, которой Fedora 18 понравится. Им не приходится скучать по проприетарным мультимедиа-кодекам, а администраторы получат множество инструментов приспосабливать новую версию под существующую сеть.

Начинающие пользователи Fedora 18 смогут добавить рабочие машины в домен Active Directoty, затем входить в систему с полученными реквизитами – благодаря интеграции в эту версию *Samba 4*, которая содержит свободную и открытую реализацию протокола взаимодействия с AD. В пользовательские настройки Gnome также включена поддержка корпоративных учетных записей предприятия.

По умолчанию установлен клиент MDSN, дающий возможность поиска таких устройств, как общие сетевые принтеры. Есть несколько улучшений в виртуализационном стеке Fedora. Сейчас можно делать снимки состояния виртуальных машин, где образ диска используется в RAW-формате. Виртуальные машины *KVM* можно приостанавливать и усыплять даже с хоста с помощью *virsh* – утилиты, запускаемой из командной строки. Версия также содержит основные компоненты *Eucalyptus* для создания и управления частными инфраструктурными компонентами, совместимыми с Amazon Web Service.

Давно уж мы не бывали так разочарованы. Приятно видеть, что Fedora опробует новинки, но делать это наобум не к лицу проекту с солидной репутацией. LXF

LINUX FORMAT Вердикт

Fedora 18

Разработчик: Fedora Project
Сайт: www.fedoraproject.org
Лицензия: различные виды свободных лицензий на ПО

Функциональность	6/10
Производительность	7/10
Простота использования	4/10
Документация	8/10

» Не советуем Fedora 18 новичкам в Linux. А новый инсталлятор создает трудности при работе с несколькими ОС.

Рейтинг 6/10

XBMC 12 (Frodo)

Бен Эверард просиживает перед телевизором гораздо дольше, чем полезно для здоровья, исследуя медиа-плейер, навеянный «Властелином колец».

Вкратце

» **Программный медиа-центр. См. также:** *VLC*, *Gnome Media Player*, *Kaffeine* **или** *MPlayer*.

Есть два знаменитых медиа-плейера свободного ПО с именами-акронимами, утратившими смысл: *XBMC* и *VLC*. *VLC* – настольное приложение для воспроизведения фильмов и музыки (а не клиент Video LAN, согласно обозначению), а *XBMC* (кхе, Xbox Media Centre) – среда для воспроизведения медиа (музыка, фильмы, прогнозы погоды, фотографии и почти все, для чего кому-то не лень написать модуль расширения). Она работает в полноэкранном режиме и предназначена для домашнего просмотра ТВ, а также монитора. Чтобы в гостиной легче жилось, вы можете управлять им с помощью устройства Android или iOS.

Впервые полная версия *XBMC* стала доступна для Android, но пока она адекватно протестирована только для Pivos XIOS DS на базе Android (мы рассмотрим его через месяц). Интерфейс пользователя приятный и легко управляется издали – это жизненно важные функции для ПО медиа-центра. Тем не менее, кое-что в использовании нам досадило. Всплывающие уведомления прячут кнопки навигации, а структура меню способна запутать. Впрочем, видали мы проблемы и похуже.

О качестве медиа-плейера говорят исполняемые на нем произведения. И здесь, пожалуй, XBMC заметно уступает коммерческим услугам: немало новинок ТВ-сервисов не интегрировано, как и платных сервисов типа Netflix. Есть обходной путь – открыть web-браузер и использовать эти услуги через HTML при помощи *Advanced Launcher* (чтобы это заработало, нам пришлось взять версию 1.10 с сайта sourceforge.net / projects / storage, а не из репозитория). Но такое отнюдь не красит медиа-центр. Подчеркнем, что виноваты в отсутствии интеграции медиа-компании – они придерживаются несвободных технологий, которые нельзя включать в системы типа *XBMC*. Если они решат уважать свободу своих пользователей, самый прямой путь – интегрировать эти технологии в XBMC и другие медиа-плейеры.

› **Прилепите Raspberry Pi скотчем к задней части старой ЭЛТ, установите *XBMC* — и вот вам крутое ТВ на £670 дешевле, чем последнее предложение компании Samsung.**

Бесплатные фильмы

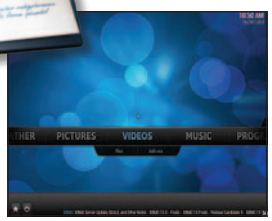
Есть множество расширений, дающих доступ к бесплатным видео онлайн. Поклонники YouTube имеют свое, а любители кино старой школы могут добыть фильмы с истекшим сроком авторских прав через расширение Classic Cinema. Вы найдете целые залежи на www.xbmchub.com. Имейте в виду, что некоторые из них используют незаконные источники контента – *XBMC* имеет расширения для управления торрентами (хотя ни Linux Format, ни автор такого не одобряют, и если вы на это пойдете, правовые последствия могут быть суровыми).

Другой вариант – использовать *XBMC* как интерфейс к системе PVR вроде MythBox (ее полная интеграция – новинка версии 12). Его нелегко настроить, но наградой станет система в стиле Tivo, которая работает со многими станциями наземного и спутникового вещания, не навлекая на себя гнев Ричарда Столлмена. Вики *XBMC* сообщит вам подробности: http://wiki.xbmc.org/index.php?title=PVR.

Важно понимать, чем XBMC является (медиа-плейером) и чем не является (источником информации). В коммерческом мире грань между этими двумя все более стирается, поскольку продавцы и ПО, и «железа» норовят выжать больше денег из клиентов, продавая контент вместе со средствами его воспроизведения.

Если нужен способ проиграть бесплатный или коммерческий контент в развлекательном центре, *XBMC*, наверно, лучший вариант. Но способом легального доступа к онлайн-медиа, защищенным авторскими правами, остается коммерческое ПО. LXF

Свойства навскидку



Интерфейс
Крупноблочный интерфейс пользователя даст вам сидеть на диване с кульком попкорна в руке.

Дополнения
Расширения существуют для всего, в диапазоне от BitTorrent до эмулятора консоли.

LINUX FORMAT Вердикт

XBMC 12 (Frodo)

Разработчик: Команда XBMC
Сайт: xbmc.org
Лицензия: GPLv2

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

» Лучший медиа-центр с открытым исходным кодом еще немного похорошел.

Рейтинг 8/10

CuBox

Грэм Моррисон разглядывает крошечную черную коробочку, где умещается целая Linux-система.

Вкратце

» Мини-компьютер ARM с высокой подключаемостью. См. также: Tonido-Plug, Raspberry Pi и медиа-плейер Pivos Xios DS.

Сейчас развелось невероятно много крошечных встраиваемых Linux-устройств, но CuBox выделяется хотя бы тем, что на миллиметр в нем приходится больше соединений, чем в любом другом. Обычно мы не перечисляем их полным списком, но на 5,5-сантиметровой задней панели Cubox помещаются HDMI-коннектор с разрешением 1080 пикселей, два скоростных USB-порта, порт Ethernet, слот MicroSD, интерфейс eSATA, разъем питания и порт MicroUSB. Прибавьте к этому оптический аудиоконнектор сбоку и инфракрасный порт спереди, и вот вам самое подробное описание 127 кубических миллиметров в наших обзорах.

Внутри у него ARMv7 с частотой до 800 МГц, 1 ГБ ОЗУ DDR (доступны также версии с 2 ГБ); на нашем была предустановлена Ubuntu 10.10 и MicroSD-карта на 4 ГБ, хотя есть и официальный вариант с Android. Поскольку Ubuntu 10.10 довольно старая, первым делом мы обновились до Ubuntu 12.10. Поскольку CuBox считается платформой разработки, еще одна ее важная функция – то, что она якобы «неубиваемая», то есть как далеко ни зашли бы ваши низкоуровневые эксперименты, способ восстановить систему есть всегда – а именно, при помощи последовательного USB-порта и эмулятора командной строки на другой машине. Таким же образом можно установить другую ОС, предварительно скачав установщик, загрузив его на флэшку перед запуском CuBox и подключившись к последовательному терминалу. Такой способ установки дистрибутива кажется чересчур техническим, но это не так уж сложно, если делать все по инструкциям с wiki.

› По картинке не понять, но ширина задней панели CuBox всего 5,5 см.

› Минималистскую цельность черной передней панели нарушают только инфракрасный порт и оптический аудиовыход.

Включай и работай

Для начала работы нужно всего лишь все подключить и нажать кнопку. В случае успешной загрузки, на передней панели загорается слегка мерцающий огонек; другой, красный, на задней панели, указывает на наличие подключения к сети. Через несколько секунд вы в рабочем столе. Вот только ни один из наших стандартных мониторов не заработал.

Как выяснилось, CuBox довольно требователен в отношении типов мониторов и разрешений экрана. По умолчанию у него блистательные 1920×1080, и это прекрасно, если у вас есть телевизор с таким разрешением, но бессмысленно со стандартным дисплеем. Немного с этим помучившись и так ничего и не добившись, мы просто воткнули CuBox в соответствующий телевизор. К счастью, SSH запущен по умолчанию, так что проблемы можно решать удаленно, но их причина так и осталась для нас неясной. Похоже, что видеооборудование на CuBox имеет какие-то странные ограничения.

Поскольку с особой гордостью в спецификациях отмечалось ускорение воспроизведения видео, мы испытали CuBox потоковым телевидением высокой четкости с нашего Raspberry Pi. Все работало хорошо, и хотя *XBMC* был не так проворен, как хотелось бы, CuBox действительно демонстрирует неплохой результат, особенно при наличии прямого доступа к источнику, минуя интерфейс пользователя.

Но несмотря на характеристики медиа, мы считаем, что это не главный козырь CuBox. Нам показалось, что он скорее похож на универсальный инструмент для разработчика, вроде Raspberry Pi, но с лучшими спецификациями и в корпусе. Его можно использовать в домашних условиях, как аудиопроигрыватель, начинку робота или локальный web-сервер. Все достоинства Linux, но в компактной упаковке. За это можно только похвалить. LXF

LINUX FORMAT Вердикт

CuBox

Разработчик: SolidRun/New It
Сайт: www.newit.co.uk
Цена: £105

Функциональность	9/10
Производительность	7/10
Удобство использования	7/10
Оправданность цены	8/10

» Еще одна чудесная миниатюрная машина Linux, идеальная для самых разных проектов.

Рейтинг 8/10

LibreOffice 4.0

Эндрю Грегори оторвался от репортажей о регби, чтобы поработать в офисном комплекте № 1 для Linux.

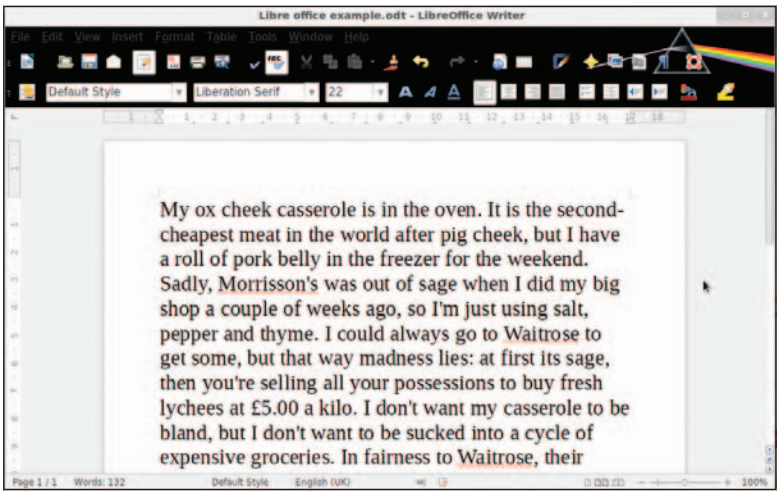
Вкратце

» **Самое многофункциональное офисное ПО для Linux. См. также:** *OpenOffice.org*, *Gnumeric*, *Calligra*, *AbiWord*.

С сентября 2010 года, когда *LibreOffice* отпочковался от *OpenOffice.org*, он превзошел все бесплатные пакеты офисных программ. Конечно, он стоит на плечах гиганта – *OpenOffice.org*, но за достижения этой версии разработчики заслуживают большого уважения.

Как говорит Майкл Микс [Michael Meeks], случайный «прихожанин» проекта, около трети нововведений пришла от Red Hat, треть от SUSE и треть от волонтеров, вклад которых резко увеличился после отделения от *OpenOffice.org* (а по сути – от Oracle) – сей факт все еще влияет на желание подарить проекту программный код.

Несмотря на всю проведенную работу, в *LibreOffice 4.0* не так много функций, которыми можно кичиться, но это говорит о зрелости продукта, а не о небрежности команды Document Foundation. Сейчас мы наблюдаем взлет новой блестящей идеи: выпуска отдельного приложения для Android, через которое можно управлять презентациями *Impress* с мобильного телефона. Майкл поведал, что следующим пунктом в списке пользовательских хотелок может стать эмулятор лазерной указки, работающий от акселерометра, встроенного в телефон, через соединение Bluetooth, который в ходе презентации будет «излучать» красную точку на экран. В команду уже поступило предложение о готовности работать над ним от студента 'Google Summer of Code'.



› **Для персонализации внешнего вида офисного пакета посетите Tools > Options > Personalisation и выберите одну из тем, созданных пользователями *Firefox* (Personas).**

Одно из новшеств версии 4.0 – Personas, продвинутая система изменения оформления, уже некоторое время существующая в *Firefox* и позволяющая в офисных приложениях перенацеливать любительское художество волонтеров на браузер. Сперва это воспринималось как шутка, но оказалось, что данную систему ввели по особому запросу – одна компания-заказчик пожелала визуально подчеркнуть корпоративный дух в используемом ПО.

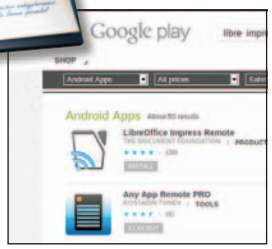
Улучшенная совместимость

Большие шаги вперед сделаны в менее броской, но бесспорно более важной области совместимости с форматами файлов других офисных продуктов. Улучшена поддержка RTF и DOCX, в версии *LibreOffice 4.0* добавлена поддержка файлов формата *Microsoft Publisher*, и это очень круто: бесплатное ПО впервые реализует такую поддержку.

Publisher – относительно примитивное настольное издательское приложение с акцентом на дизайн, и есть шансы, что *LibreOffice* выбьется в конкуренты такому инструменту верстки, как *Scribus*; но пока его цель – увеличить потенциал приложений, вобрать информацию, а потом уж придумывать, что с ней делать дальше. Подобная философия, хоть и преследует далеко идущие цели, при тестировании выигрышно не смотрится. Но зато именно такое и требуется от офисного пакета, особенно столь важного для экосистемы свободного ПО. Если вам пригодятся дополнительные функции этой версии, то и замечательно. А для кого-то они будут только лишним уровнем в меню при очередном поиске функции.

Быстрее ли новая версия, чем 3.x? Если честно, не ощутили, но те, кому скорость главнее, выбирают *AbiWord*, а то и *Nano*. А вот что имеет значение – то, что новая версия стала элегантнее, больше умеет и лучше ладит с проприетарными форматами Microsoft; а значит, пригодится более широкому кругу людей. LXF

Свойства навскидку



Дистанционное управление

Загрузите приложение на мобильник с Android и используйте его для дистанционного управления презентациями *Impress* через Bluetooth.

Интеграция с Unity

Приложения любимого свободного офисного комплекта теперь плавно интегрируются со свободным ПО ~~любимее~~ четвертого по популярности дистрибутива.

LINUX FORMAT Вердикт

LibreOffice 4.0

Разработчик: The Document Foundation
Сайт: www.libreoffice.org
Лицензия: GNU LGPL v3

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

» Новшества прячутся внутри, подобно фундаменту мощной крепости. Важны для совместимости.

Рейтинг 9/10

Legend of Grimrock

Сорвавшись с мрачного утеса, **Евгений Балдин** сражается за выживание в пестрой компании магов, монстров, инсектоидов и не только.

Вкратце

» Старая добрая бродилка от первого лица по подземным лабиринтам — но на новый лад.

Четыре независимых финских разработчика игр, сбившись в компанию Almost Human, взялись за вроде бы неподъемную задачу: перенести для возмужавшей аудитории детские воспоминания об играх *Eye of the Beholder* и *Ultima Underworld* в современный мир красивых картинок. И почти все элементы захватывающего изучения подземелий сложились в их проекте весьма удачно. Есть, конечно, упрощения – так, в помещении размером две на две клетки вы партией любого уровня победите фактически любого противника, если тот один. Но даже такие послабления пошаговой механике игры лишь добавляют ей шарма.

После успешного старта весной 2012 г. обделенные пользователи GNU/Linux умоляли портировать игру. Финны помалкивали, но когда надежда почти умерла, стартовал Humble Indie Bundle 7, где *Legend of Grimrock* была приманкой тем, кто заплатит выше среднего. За 2012 г. на поле GNU/Linux вышло много славных игроков, а будет их и больше, но всегда надо помнить о Humble Indie Bundle – как о предтече.

Порт под GNU/Linux не лишен недостатков. Игра проходится без особых проблем, но такая мелочь, как отсутствие ввода текста, препятствует бонусному уровню за Toorum'a – супербойца, чьи ободряющие записки обнаруживаются в самых разных уголках этого угрюмого мира. Финские парни в курсе проблемы и сулят ее исправить. Понадеемся на их обязательность.

Legend of Grimrock – классическая бродилка от первого лица по подземному лабиринту [dungeon crawl]. На вершину зловещей горы (Grimrock) аэростатом доставляют особо провинившихся заключенных и сталкивают их в пропасть, которая ведет на верхний уровень подземелий. Оттуда только один путь – вниз!

После несложной заставки вам предлагается сколотить партию приключенцев – они будут действовать как единое целое. На выбор дают 4 расы: люди, минотавры, ящеры и инсектоиды, и 3 класса: воин, маг и вор. У каждой из рас свои преимущества и недостатки. Для силовой поддержки партии и вора-лучника годятся минотавры – в обоих случаях нужна лишь сила. Магу лучше придать инсектоида. Брать двух магов не очень умно. Магия, конечно, аргумент весомый, но заклинать надо в реальном времени, каждый раз набирая должную комбинацию рун. Простой топор или арбалет здесь эффективнее: на них для атаки достаточно щелчка. Но персонажей можно развивать, что поощряет к созданию экспериментальных партий – скажем, из одних магов. С броней у них слабо, но можно развить ветку защиты Staff Defense, и мало кто дорвется до партии, получив в морду четыре огненных шара. Герои строятся в две шеренги. В первой ставьте персонажей с крепким здоровьем и хорошей защитой: им перепадут почти все удары. Вторая защищена от фронтальной атаки и может заняться магией или луком. Но при атаке с тыла вторая линия погибнет быстро; будьте начеку.

› Этот монстр не только сильный маг, но умеет и открывать двери! В душе он белый и пушистый... правда, любовь к разумным существам выражает странновато.

Для выживания нужна пища. Сперва с ней все хорошо, но позже, особенно если темп исследования нетороплив, ее уже не хватает – делайте запасы. Простое правило – съесть тяжелую пищу раньше легкой, особенно это касается мяса улиток и кусочков разумных грибов. Иначе никакой минотавр не унесет собранные харчи, а аппетит у него больше, чем у всех.

Двери и решетки – ваши друзья: чудовищам, кроме самых разумных, их не открыть. Поэтому тактика «открыл решетку, выстрелил в зверя и тут же закрыл, пока зверь не вдарил в ответ» действует на ура, особенно у лучников, ведь стрелы попадают во врага всегда, сбивая его с атаки.

Почти каждый из 13 уровней подземелий нашпигован секретами и загадками. Восьмой уровень вообще можно проскочить, если внимательно не прислушаться, нажав секретную кнопку на седьмом. В процессе решения некоторых загадок с участием телепортов и движущихся полов временами возникает желание расшибить клавиатуру или монитор: клавиатурные комбинации любого из файтингов – детский лепет по сравнению с последовательностями, которые приходится вытворять героям. С другой стороны, в этом и кроется немалая доля привлекательности данного приключения.

Сюжет подобным бродилкам не особо нужен, достаточно цели – выжить. Но есть пара сюрпризов от разработчиков. Мир под горой не так тривиален, а число и сила монстров зашкаливает за все пределы по Причине, которую знает Главгад. LXF

LINUX FORMAT Вердикт

Legend of Grimrock

Разработчик: Almost Human
Сайт: http://www.grimrock.net/
Цена: $14,99 через Ubuntu Software Center

Сюжет	6/10
Графика	7/10
Увлекательность	10/10
Оправданность цены	8/10

» Шедевр, а для шедевра сюжет и графика не самое главное, хотя я, возможно, пристрастен.

Рейтинг 8/10

PocketBook Touch

Артем Зорин почитал, посмотрел и подивился классической читалке.

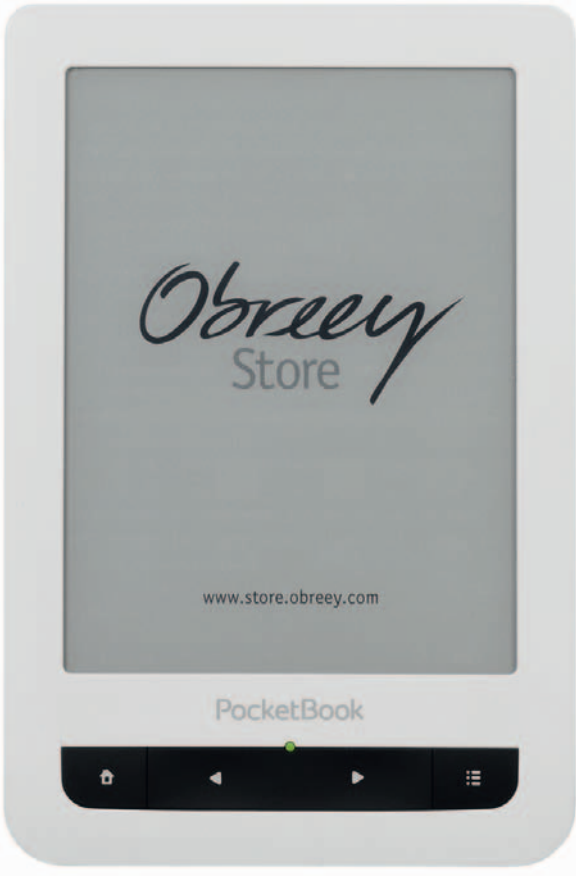
Вкратце

» Настоящая, классическая электронная книга — такая, какой она и должна быть.

В начале 2013 года трудно кого-либо удивить электронной книгой – ими уже пользуются, как обычными, бумажными; они есть у многих. Однако есть среди всего многообразия читалок такие, которые «цепляют». Одна из них – PocketBook Touch 622. Это устройство для тех, кто любит читать и читает. Идеально для летнего чтения в гамаке в саду. Качество экрана вне конкуренции. Дизайн – изумительный, высший класс. А главное, что «цепляет» – это потрясающий экран.

Не будем вдаваться в скучные «техникалии»; кому это интересно – загляните на официальный сайт PocketBook: http://www.pocketbook-int.com/ru/products/pocketbook-touch. Скажем о главном: процессор FreeScale на 800 Мгц и 128 МБ ОЗУ, вкупе с отличным аккумулятором на 1100 мАч, позволяют перелистывать страницы в 2,2 раза быстрее, чем на предыдущих моделях (по словам производителя), а заряда батарейки хватит на 8000 перелистываний, и это *очень* много. Раз уж основная функция читалки – чтение книг, этого более чем достаточно. Электронной книге нет нужды играть мышцами, в отличие от планшетов.

› Любителям почитать, и просто ценителям красивых гаджетов — вот для кого этот ридер.



Из прочих «плюшек» отметим наличие Wi-Fi 802.11 b/g/n, возможность слушать музыку через наушники и 2 ГБ встроенной памяти для книжек, расширяемой до 34, если вставить карточку microSD на 32 ГБ; представьте, сколько литературы можно таскать с собой – целую библиотеку!

Все слоты и кнопки – разъем для подключения наушников 3,5 мм, слот для карты microSD, порт microUSB, разъем сброса настроек и форматирования памяти, кнопка Вкл/Выкл – вынесены на нижний торец, что тоже разумно – по бокам нет ничего лишнего. Пластик мягкий и приятный на ощупь, не скользит. Книжка удобно помещается в кармане – ведь экран в 6 дюймов не столь велик, но вполне удобен для чтения.

Multitouch-экран и технология E-Ink Pearl делают свое дело на 5 баллов – реакция на нажатия быстрая, тормозов не замечено. Об экране, кстати, еще раз упомянем чуть ниже.

Фирменная технология PocketBook под названием «Text To Speech», переводящая текст в русскую речь, работает на удивление хорошо, практически без заминок и коверканий великого и могучего, что тоже весьма порадовало.

Казалось бы, обычная, даже немного морально устаревшая читалка. Так что же в ней такого, чем она «цепляет»? Очень просто – отличный экран. Он идеален для чтения вне дома, то есть не в помещении. Нормальные люди не пользуются читалками в помещении – для этого существуют планшеты, смартфоны, ноутбуки, просто книги, в конце концов. Все эти TFT, OLED, SuperAMOLED, Retina и прочие ЖК-дисплеи пока хороши лишь при неярком домашнем свете. Стоит вам оказаться на улице – равных экранам с электронными чернилами нет, они вне конкуренции на дневном солнечном свету.

Это нишевое устройство для чтения на открытом воздухе. Страницы можно перелистывать как двигая пальцами по экрану, так и кнопками навигации внизу. Все это работает быстро. Конечно, язык не поворачивается сказать, что книжка дешевая – 6490 руб, ведь есть уже и за 3000 и даже меньше, но это – PocketBook, именитый производитель, зарекомендовавший себя на рынке электронных книг только с положительной стороны.

Пройдемся по софту. Он стандартен для читалок: есть собственно программа для основной функции – чтения – «Библиотека», музыкальный плейер, англо-русский словарь ABBYY Lingvo, встроенный магазин литературы и музыки Obreey Store, программа для заметок от руки *Scribble*, web-браузер (!), калькулятор, менеджер фотографий с возможностью импорта снимков на компьютер и с него, часы-календарь, RSS-читалка и три игры: Шахматы, Судоку и Косынка. Все работает шустро и без тормозов. При первом подключении к сети читалка предлагает скачать и установить обновление прошивки размером около 150 МБ, и после обновления все начинает работать еще быстрее и лучше. И да, работает все это на ядре Linux 2.6, что немаловажно – можно попробовать перепрошить книгу альтернативной прошивкой, но на свой страх и риск и с потерей гарантии, которая у PocketBook довольно продолжительная – от года до трех лет.

Имеется уже целая прорва книг – классическая русская и иностранная литература. «Ест» PocketBook Touch почти все подряд – форматы книг EPUB DRM, EPUB, PDF DRM, PDF, FB2, FB2.ZIP, TXT, DJVU, HTM, HTML, DOC, DOCX, RTF, CHM, TCR, PRC (MOBI), картинки форматов JPEG, BMP, PNG, TIFF и музыку в MP3.

Экран, производительность, изумительный дизайн от Obreey Products, потрясающе долгая работа от батареи и весь необходимый софт делают PocketBook Touch идеальным подарком – и спутником любителя почитать. Однозначно: надо брать. LXF

LINUX FORMAT Вердикт

PocketBook Touch

Разработчик: PocketBook International S.A.
Сайт: http://www.pocketbook-int.com/ru/products/pocketbook-touch
Цена: от 6490 руб.

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

» Хорошая, добротная, удобная электронная книга — и отличный подарок.

Рейтинг 10/10

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.

ПОЛУГОДОВАЯ ПОДПИСКА 1230 РУБ.

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

ТЕЛЕФОН В КОСМОСЕ

Система запущена раньше NASA

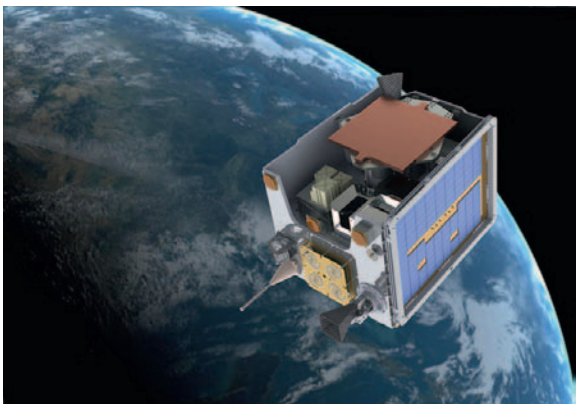
Англичанам удалось-таки запустить Nexus One на околоземную орбиту.

Ранее мы уже сообщали о проекте NASA по созданию нового поколения маленьких, дешевых космических аппаратов, построенных на базе телефонов с Android и названных PhoneSat. Проект PhoneSat все еще находится на стадии подготовки к полету, а британские ученые сумели опередить американских коллег, запустив на околоземную орбиту искуственный спутник под названием Surrey Training, Research and Nanosatellite Demonstrator (STRaND-1).

STRaND-1 был успешно запущен 25 февраля 2013 года с Космического центра имени Сатиша Дхавана [Satish Dhawan Space Centre] (остров Шрихарихота, Индия), и, если не случится ничего не предвиденного учеными, спутник переключится на управление интегрированным в его корпус смартфоном Nexus One уже через несколько дней. Волне возможно, что во время выхода этого номера в печать STRaND-1 уже работает под управлением Android.

STRaND-1 – очень интересный в своем роде космический аппарат, даже если игнорировать тот факт, что это первый спутник на основе смартфона, летающий в космосе (хотя и не первый смартфон, побывавший в космосе).

Интересен двигатель спутника, основанный на самых последних достижениях инженерии: в качестве основного аппарат использует так называемый водно-спиртовой омический импульсно-тормозной возвратно-скоростной экспериментальный двигатель – Water-Alcohol Resistojet Propulsion Deorbit Re-entry Velocity Experiment (WARP DriVE – тут для понимающих прослеживается некоторая аллегория со знаменитым сериалом Star Trek, в котором звездолет Enterprise использовал двигатели искривления пространства, называемые WARP Engines, для набора сверхсветовой скорости); также используются 8 импульсно-плазменных двигателей. Эти системы разработаны, чтобы помочь спутнику маневрировать на орбите. Также в спутнике используются компоненты, напечатанные на 3D-принтере. Создатели даже называют его первым в мире спутником, почти полностью напечатанным на 3D-принтере, исключая сердце – смартфон Nexus One.



Размером с небольшую коробку со смартфоном и весом всего 3,5 кг, STRaND-1 способен на многое.

В дополнение к Nexus One, STRaND-1 также укомплектован специально разработанным маленьким компьютером на основе особой, «космической» версии Linux. Именно этот компьютер управляет в данный момент спутником, пока команда создателей готовится переключить его на управление смартфоном.

Использование уже устаревшего ныне Nexus One объясняется тем, что проектирование и разработка STRaND-1 началась более трех лет тому назад, и первый Nexus был тогда достаточно современным.

> «Британские ученые сумели опередить американских коллег.»

По словам создателей, они предпочитают пусть и немного устаревшие морально, но проверенные и надежные инструменты, напоминая, что некоторые космические челноки NASA, летавшие в Космос всего несколько лет назад, до последнего управлялись компьютерами на 386-х процессорах...

Но больше всего в STRaND-1 впечатляет «чистый» Nexus One на его борту, то есть никак не модифицированный и нисколько не измененный, как будто он только что был куплен в магазине. В отличие от англичан, NASA планирует применить в «космическом» смартфоне спецпрошивку.

STRaND-1 представляет собой обычный «стоковый» телефон без модификаций, облаченный в специальный прочный корпус и оснащенный дополнительным компьютером, и его задача – показать, сможет ли обычный смартфон «выжить» в суровых космических условиях, используя лишь надежность и гибкость платформы Android и Linux. Проблему переохлаждения батареи телефона в открытом космосе планируется решать так: бортовой компьютер под управлением Linux будет следить за температурой аккумулятора телефона. Если сенсоры компьютера зафиксируют снижение температуры батареи ниже определенной отметки, в смартфоне автоматически включится программа, интенсивно использующая процессор и видеочип телефона, что должно разогреть аккумулятор.

Как только команда создателей закончит анализ и тестирование всех систем спутника, а также смартфона, и убедится, что телефон может нормально функционировать, они передадут управление основными системами космического аппарата непосредственно Nexus One. После получения контроля над спутником он будет использовать ряд стандартных приложений Android для получения изображений Земли, измерения звуковых и электромагнитных волн Планеты и передачи данных телеметрии на Землю.

«STRaND-1 ломает многие стереотипы, связанные с космическими технологиями, и если все пойдет по плану, он станет началом долгого пути роста гражданских частных и коммерческих космических миниспрограмм. Если он сможет доказать, что обычный смартфон под управлением обычного Android может приносить пользу, и делать сложную научную работу, это будет начало новой эры в космических исследованиях», говорят создатели спутника.

Отметим, что NASA запустит целых три спутника PhoneSat уже в этом месяце. Это будет сделано с помощью новой научной орбитальной ракеты-носителя Antares – последнего слова техники в коммерческих гражданских космических аппаратов. В наноспутниках PhoneSat будут использованы телефоны Nexus One и Nexus S.

Всю дополнительную информацию по проекту см. на сайте www.sstl.co.uk.

Фото © www.sstl.co.uk

ОПАСЕНИЯ ПОДНЕБЕСНОЙ

Чем Google помешал китайским производителям?

Власти Китая обеспокоены господством Android на мобильном рынке страны.

Власти Китая дали понять, что хотят уменьшить свою зависимость от Android, мотивируя это тем, что Google дискриминирует местных разработчиков, не давая им развивать свои мобильные платформы.

«Исследования и развитие мобильных платформ в нашей стране сильно зависят от Android. Хотя основные компоненты Android и являются открытыми, некоторые технологии и решения, а также планы развития платформы находятся под строгим контролем Google», говорят в Министерстве промышленности информационных технологий КНР.

По итогам исследования рынка мобильного Интернета, проведенного китайскими властями, пока не предлагается никаких конкретных мер против Google, однако они отметили доминирование Android на китайском рынке смартфонов, хотя все чаще китайские производители стараются использовать исключительно отечественные технологии.

Предприятия Китая, развивающие свои мобильные платформы, время от времени сталкиваются с так называемой «коммерческой дискриминацией» со стороны Google, говорится в исследовании. Предполагается, что Google намеренно задерживает открытие исходных кодов новых версий Android, а также часто использует коммерческие соглашения, запрещающие использование некоторых частей исходного кода Android, что ограничивает и сдерживает развитие мобильных технологий в Китае. В исследовании особо подчеркнуто, что разработка основных компонентов, технологий и патентов находится в руках американских и европейских компаний.

В Google, у которой всегда были непростые отношения с китайскими властями, отказались что-либо комментировать. Напомним, что в 2010 году компания частично перевела свой бизнес из Поднебесной в Гонконг, после скандала с кибер-атаками на сервера Google с китайской стороны и отказа вводить на сайтах Google цензуру и слежку за пользователями по требованию властей Китая.

Пока неясно, какие именно китайские фирмы столкнулись с предполагаемой дискриминацией со стороны Google. Однако в прошлом году гигант китайского рынка электронной коммерции Alibaba Group уже судился с Google из-за своей мобильной Linux-подобной платформы Aliyun OS, которая имела поразительное сходство с Android как внешне, так и внутренне. Спор закончился полным запретом продаж смартфонов с Aliyun OS где бы то ни было, и запуск новой системы был сорван.

› Судя по взрывному развитию в Китае рынка всего, у альтернативы Android есть будущее.

> «Google сдерживает развитие мобильных технологий в Китае.»

Министерство промышленности и информационных технологий Китая также отказалось от комментариев. «Последняя новость была расценена всеми участниками рынка как четкий сигнал властей Китая всем заинтересованным лицам об их твердом намерении поддерживать свою, китайскую ИТ-индустрию. Я думаю, что это очень последовательная и долгосрочная политика китайского государства по содействию развитию отечественных стандартов и технологий, чтобы быть менее зависимым от зарубежных компаний», отметил Марк Наткин [Mark Natkin], исполнительный директор пекинской консалтинговой фирмы Consulting Marbridge.

«Наша страна уже применяет такую политику в сфере телекоммуникаций. В 2009 году государственный мобильный оператор China Mobile запустил собственную 3G-сеть с использованием исключительно отечественных технологий, под названием TD-SCDMA [Time Division Synchronous CDMA]. Хотя технология и не получила распространения за пределами Китая, она дала сильный толчок развитию сетей четвертого поколения в Китае. Политика протекции отечественных производителей мобильных платформ рассчитана на длительный неопределенный срок, и вовсе не обязательно, чтобы она приносила свои плоды уже сегодня», добавил Наткин.

В то же время, китайские производители мобильных телефонов, такие как Huawei, ZTE и Lenovo, успешно использовали и используют Android в своих устройствах, чтобы продавать их не только на территории Китая, но и за его пределами. По данным китайской консалтинговой фирмы Canalys, в четвертом квартале 2012 года три вышеназванные компании заняли в общемировом рейтинге продаж мобильных устройств третье, четвертое и пятое места соответственно, лишь немного уступив своим американским и южнокорейским конкурентам.

В Правительстве Китая говорят, что их страна уже создала «хороший фундамент» для развития собственной, независимой отрасли мобильных технологий, указывая на положительный опыт Huawei, Alibaba и Baidu, которые сделали мобильный бизнес ключевым элементом своей стратегии.

Российским властям есть чему поучиться у своих китайских коллег. LXF

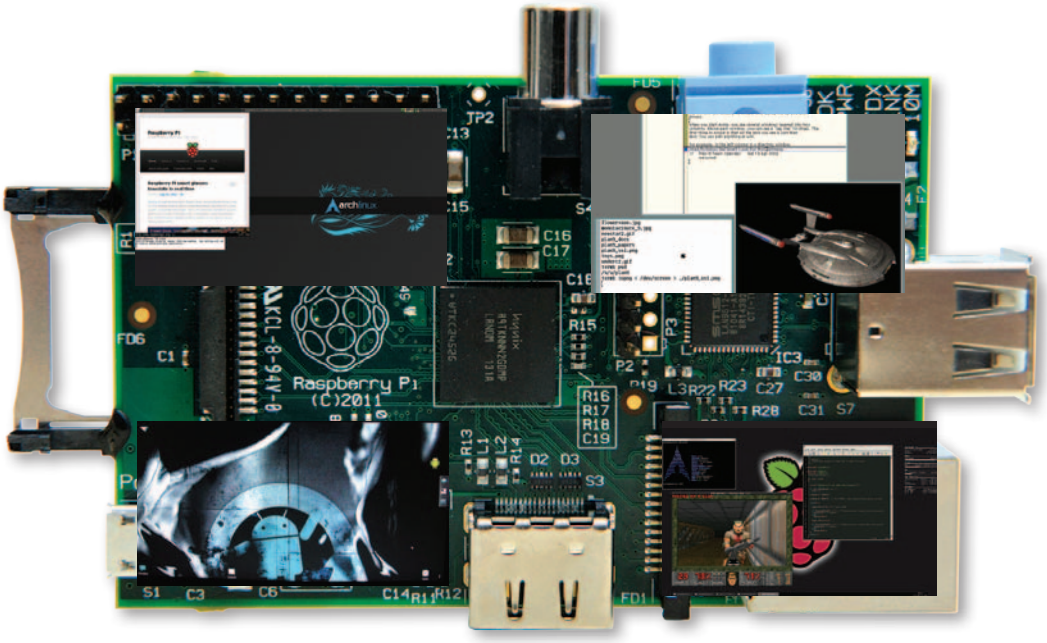
Фото © http://media.tumblr.com. В данном выпуске новостей использованы материалы с сайтов www.thepowerbase.com и www.it-world.com

Сравнение

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

Операционки для Raspberry Pi

Дэвид Хейвард рассматривает со вкусом подобранные ОС для Raspberry Pi.



Про наш тест...

Raspberry Pi сейчас выпускается в двух вариантах: исходная модель B версии 1 с памятью 256 МБ, и более новая модель B версии 2 с 512 МБ.

Поэтому для поистине всеобъемлющего обзора мы не пожалели времени на установку операционных систем на SD-карту объемом 4 ГБ и для новой модели с 512 MB, и для старой с 256 MB. Мы рассматривали такие аспекты, как установка; программы по умолчанию; воспроизведение мультимедиа (немедленное); внешний вид и удобство в использовании; сообщество, поддерживающее ОС, и, соответственно, его отношение к свободе ПО. То есть те самые факторы, которые и определяют выбор пользователем операционной системы.

Мы хотели рассмотреть все эти факторы с точки зрения человека, не слишком знакомого с Linux, чтобы такой пользователь смог без особой мороки начать работу с проектом и не опустил бы рук в разочаровании.

Наша подборка

» Android
» Arch
» Plan 9
» Raspbian
» Risc OS

Похоже, феномен Raspberry Pi набирает обороты; словно поезд, слетевший с тормозов, он мчится вперед, побивая все рекорды. Удивляться тут нечему: оборудование разработано в идеальном соответствии с целями Raspberry Pi Foundation, ценовая политика выверена идеально, и все это подкрепляется многогранностью Linux как операционной системы.

Большинство покупателей, заполучив новенький RPi, направляются на официальный сайт Raspberry Pi и следуют предлагаемым там инструкциям; в итоге они послушно обращаются к Raspbian “Wheezy”, операционной системе, рекомендованной Foundation, что-то создают, учатся и программируют, запихивают несчастного зверька в метеозонд и отправляют его в дальние слои атмосферы. Однако многие пользователи RPi не осознают, что ОС для их любимца полно.

> «Многие пользователи RPi не осознают, что ОС для их любимца полно.»

И мы решили, что донесем до несведущих пользователей информацию об этих восхитительных топпингах для их пирожка с малиной [Raspberry Pi – «Малиновый пирог», – прим. пер.], и приложим в подарок коробку шоколадного ассорти – тщательно продуманную подборку операционнных систем.

Установка

Надо ли тут быть доктором наук?

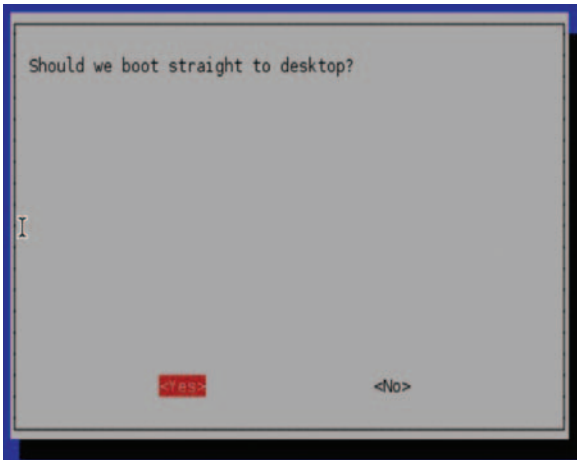
Установка образа операционной системы весьма неплохо задокументирована, в разделе сайта Raspberry Pi под названием «Руководство для начинающих [Guide for beginners]», который можно найти на goo.gl/53xgp, вместе с простой процедурой установки после переноса образа на SD-карту с помощью **dd** в Linux или *Win32DiskImager* в Windows. Перенос труда не составит, а настоящая-то потеха начинается именно после установки SD в Raspberry Pi и его включения.

Все пять операционок нашего Сравнения – Raspbian, Risc OS, Arch, Android 2.3 и Plan 9 – имеют собственные нюансы и методы, позволяющие их установить и обеспечить пользователя базовым графическим рабочим столом. Хотя наличие GUI вовсе не является обязательным условием, все же огромной доле пользователей, знакомящихся с Linux, GUI подходит больше. В подобном случае понятие «установка» должно включать возможность добраться до того момента, когда пользователь распознает ОС подобно тому, как узнают стандартный рабочий стол Linux – иными словами, перед ним должен предстать графический пользовательский интерфейс.

В мире, где главным является облегчение пользователю процесса нырка в воды Linux, Raspbian занял было верхнюю ступень пьедестала почета, но другие предложения тоже служат неплохим стартом для новичка. Возьмем, например, Risc OS; перенеся ее на SD-карту и загрузив, мы мигом оказались в красочном и дружелюбном GUI, с довольно подробными сообщениями обо всех проблемах при изначальной загрузке и настройке. Здесь можно просто нажать на значок Configure и начать менять все, что заблагорассудится.

Arch Linux для RPi ведет себя иначе: он открывает терминал и дает пользователю возможность скачать, установить и настроить свою ОС. Если оценить его по достоинству, Arch является одной из лучших операционных систем, но требует настройки, чтобы добиться от него нормального рабочего стола.

Имеющийся на данный момент Android для проекта RPi начинает обретать некую форму; после загрузки перед вами появится официальный экран разблокировки Android, а за ним – тот интерфейс, который мы все знаем и (иногда) любим. Он пока не лишен ошибок, периодически сбоит и крайне медленный – до такой степени, что на 256-МБ RPi использовать его просто невозможно; версии 512 МБ работают чуть получше.



› Raspbian некогда считался лидером по простоте установки, но конкуренты, судя по всему, уже наступают ему на пятки.

Plan 9 предоставляет эффективный GUI, который, однако, довольно труден в освоении, хоть и снабжен хорошей документацией. Raspbian после загрузки предлагает текстовое меню. Пользователи могут настроить систему, включить SSH и автоматически загружать дружелюбный к пользователю GUI *LXDE*.

Мы бы порекомендовали начинающим пользователям взять Raspbian, а затем, по мере освоения Linux, переходить на Risc OS и Arch.

Вердикт

Raspbian ★★★★★
Risc OS ★★★★☆
Android ★★★☆☆
Plan 9 ★★★☆☆
Arch ★★☆☆☆

» Raspbian ведет на знакомый GUI рабочего стола, и очень прост.

Программы по умолчанию

Чем угостить сладкоежку-пользователя Pi?

Выбор программ в каждой операционной системе для Raspberry Pi весьма специфичен и отличается от других, но именно получаемое с ходу зачастую становится определяющим моментом при выборе операционной системы.

Нам все же не стоит рассчитывать на нечто особо сложное – в конце концов, это же операционная система, которая работает с SD-карты и работу которой обеспечивает компьютер размером с кредитную карту и с минимальным (по сравнению с настольной системой) набором ресурсов.

Офисные программы и полные пакеты для работы с мультимедиа или редактирования графики являются нормой для стандартного дистрибутива, но то, что было включено в выбранные нами для Сравнения ОС для RPi нас приятно удивило.

› Нас впечатлило обнаруженное в этой категории, что и демонстрирует Arch.

Лидером по части набора программ по умолчанию является Raspbian (как и следовало ожидать), однако Risc OS следует за ним по пятам; и если вы не против доплаты £35 за NutPi Pack, вам предложат полноценный рабочий стол Raspberry Pi, дополненный офисными приложениями, интернет-браузерами, программами для обмена сообщениями и т. д. Arch, как мы уже говорили, загружает терминал, но если вы знаете команды, вы сможете добиться результата, мало чем отличающегося от настольного.

Однако поиски того, кто мог бы стать вровень с Raspbian, увели нас от темы набора программ по умолчанию.

Удивил нас Android – что было вызвано достойным набором мультимедиа-приложений и отсутствием офисных. Почему-то мы ожидали, что проект RPi Android будет небогат приложениями, и были в шоке, увидев как раз кандидатов на отсутствие в системе.

А вот Plan 9 имел бледный вид. Продравшись сквозь среду рабочего стола, Rio, мы обнаружили, что большого выбора для продуктивной работы у нас нет – особенно это заметно новичкам.

Вердикт

Raspbian ★★★★★
Plan 9 ★★★★☆
Risc OS ★★★★☆
Android ★★★☆☆
Arch ★★☆☆☆

» Если вы оплатили NutPi Pack, то выбор Risc OS был верным.

Внешний вид и удобство

Внешний вид и удобство использования – это далеко не все. Да нет, все!

Внешняя привлекательность рабочего стола остается для многих пользователей Linux этаким порочным удовольствием. Страстно убеждая мир в том, что операционная система – это просто голая машина, пользователь тут же устанавливает себе на рабочий стол вращающиеся трехмерные объекты, огонь, Conky-образные циферблаты и всякие-разные прелести и красивости. Что ж, это дело личное, и, несмотря на солидный ущерб ресурсам, мы все любим украшать свой рабочий стол.

Конечно, с определенного момента шик рабочего стола вступает в противоречие с удобством его использования – помните Vista? Поэтому, хотя любая из наших ОС для RPi обладает потенциалом для умопомрачительной красоты, им приходится идти на уступки из-за ограниченных ресурсов RPi.

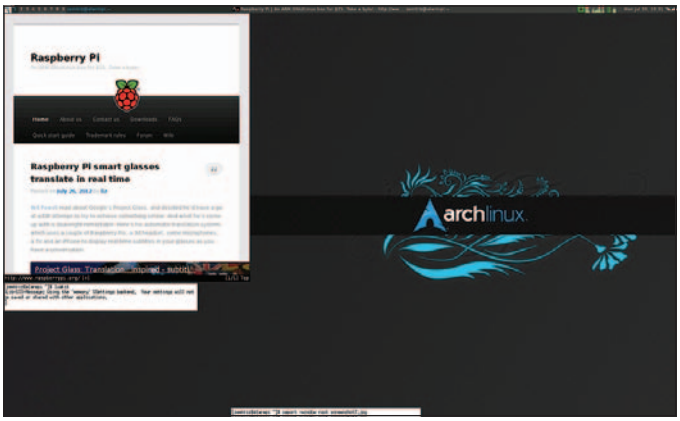
Рассмотрев все наши ОС от скромного начала и до того, чего можно от них добиться, приложив немного усилий, мы уверены, что вы сумеете найти компромисс в предлагаемой подборке.

Raspbian ★★★★★

С самого начала Raspbian предоставляет пользователю ненавязчивый, но функциональный рабочий стол. Использование *Xfce* в качестве рабочего стола означает, что ресурсы RPi находятся в надежных руках и не расходуются на пустые красивости. Однако тот факт, что Raspbian – это дистрибутив на базе Debian, означает, что рабочий стол можно весьма солидно изменить.

Можно, например, установить Mate для Raspbian, что придаст рабочему столу вид и свойства классической среды Gnome 2, и вы сможете довести настройку до полного удовольствия.

В плане удобства в использовании Raspbian изначально серьезно опережает своих собратьев, однако азартный пользователь может увлечься командой **apt** и установить сотни программ, так что его операционная система почувствует себя китом в тазике.



Arch ★★★★☆

Arch очень забавен. Он появляется на RPi, не предъявляя ничего, кроме Terminal; однако, не пожалев времени на углубленное знакомство с Arch, вы откроете одну из лучших среди существующих операционных систем.

Arch по природе своей ускоренная ОС, и даже после установки среды рабочего стола наподобие, скажем, OpenBox система будет откликаться очень быстро. Что до внешнего вида – объединив OpenBox с темой Raspberry Pi и Conky, вы твердой стопой встанете на путь к весьма привлекательному рабочему столу.

Как и внешний вид, удобство в использовании Arch целиком и полностью зависит от установленного вами, и в конечном счете это делает Arch самой настраиваемой и удобной в использовании ОС. Установка нескольких пакетов – и Arch будет работать у вас безотказно и превзойдет своего более крупного собрата, Raspbian.

Сообщество

Как в телесериале, но без насилия, склок и смертей...

Если что и выделяет Linux, превращая его в народную ОС, то это сообщество. Состоящее из отдельных индивидуумов, готовых помочь, поделиться и продвинуть свою любимую ОС, сообщество Linux – это место, где новички могут учиться, а люди опытные – дарить свои знания. Конечно, бывают здесь и тролли, и идиоты – как и везде.

У Raspbian и Arch сообщества самые обширные. Особенно у Raspbian, поскольку ему уделяется основное внимание на форуме сайта Raspberry Pi. Однако последователи Arch вполне способны соперничать с ним – и их количество растет по мере освоения новых ОС для RPi пользовательской базой.

Проект Raspberry Pi Android набирает силу, и все больше пользователей прицеляется к потенциалу этого порта для RPi; но пока он особо не блещет. Risc OS, которая завела последователей много лет назад, обладает очень ярким сообществом. Plan 9 для RPi, возможно, моложе, но у его последователей много подробных файлов онлайн-помощи, которые существуют с момента его первого релиза в 1992 году.

Большая часть предложений сообщества размещается на сайтах соответствующих ОС, однако найдите время побродить и по форуму Raspberry Pi, чтобы найти искомое. При проблемах – не стесняйтесь задать вопрос: пользователи на форуме RPi дружелюбны и более чем охотно помогают новичкам, да и продвинутым пользователям.

Вердикт

Raspbian ★★★★★
Risc OS ★★★★☆
Arch ★★★☆☆
Plan 9 ★★★☆☆
Android ★★☆☆☆

» Сообщество Linux – это ресурс, который стоит использовать и поощрять.

Android ★★★☆☆

Android 2.3 сам по себе является весьма симпатичной операционной системой. Достойная анимация, значки с высоким разрешением, анимированные обои и рабочие столы превращают ее в очень привлекательную среду. Однако при использовании на Raspberry Pi все эти прелести становятся несколько кривоватыми.

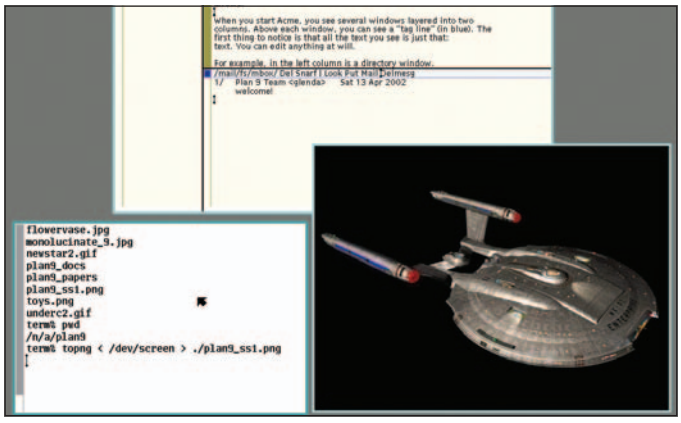
Сейчас проект Android для RPi далек от совершенства, но он еще только ожидает завершения, и постепенно превзойдет других – мы просто не уверены в сроках. В своем нынешнем виде, для работы на 256-МБ RPi он практически бесполезен, однако увеличение ОЗУ RPi до 512 МБ может слегка улучшить ситуацию. Это еще не система, пригодная для повседневного использования, но приглядывайте за этим проектом – в скором времени он может занять лидирующие позиции в новостях RPi.

Risc OS ★★★★☆

Risc OS, подобно Raspbian, при запуске демонстрирует симпатичный GUI. Risc OS GUI – хорошо продуманная и красочная среда; вдобавок очень неплохо, что эта среда загружается с разрешением 1080, при условии, что ваш монитор справляется с данным разрешением.

Что касается внешнего вида, основной рабочий стол – это, вероятно, все, что вам вообще когда-либо потребуется; он напоминает о славном прошлом Risc, придавая ему этакий налет ретро, однако в неплохом 8-битном качестве.

В плане удобства в использовании, Risc OS довольно специфична и требует наличия у вас определенного опыта работы с ней. Например, по умолчанию отключен порт Ethernet – инструкции по его включению есть, но это может отпугнуть новичка. Однако, как и в большинстве областей работы с компьютером, упорство побеждает, и мы гарантируем, что через час знакомства вы будете лихо сновать в Risc OS, как профессионал.



Plan 9 ★★☆☆☆

Если вам уже приходилось пользоваться Windows, Mac OS X, Linux или даже Unix, то, приступая к работе с Plan 9, вы смело можете похоронить все свои накопленные знания. Команды выглядят знакомыми, но разобраться в них – задачка не из легких. К сожалению, кривая обучения здесь весьма крута, что затрудняет старт, однако есть довольно значительная по объему документация.

Rio, система работы с окнами Plan 9, ощущается и смотрится как шаг назад к былым дням Amiga и Atari ST, поскольку фасад почти тот же. На привыкание к этому требуется время, и зрелище огромного количества материалов для прочтения вызывает робость, однако, как мы уже не раз повторяли, держитесь, и очень скоро вы откроете для себя симпатичный рабочий стол, простой, элегантный и летающий со свистом.

Пригодность для детей

Дружелюбны ли они к малышам?

Как мы знаем, Raspberry Pi полюбился юному поколению компьютерных всезнаек, но насколько удобны в использовании выбранные ОС?

Предлагать будущей смене инструменты для обретения опыта с компьютером очень мило, но если задача заставить железку общаться с пользователем по-человечески окажется чересчур сложной, не отпугнет ли это детей навсегда, не пропадет ли у них вообще всякий интерес? И мы попросили наших добровольцев, 11-летнего Дэниела и 10-летнюю Ханну, настроить для нас операционные системы.

Raspbian и Risc OS вышли в лидеры, оказавшись очень простыми в настройке, запуске и использовании. Android оставил у них не самое приятное впечатление, как и бедный старый Plan 9. Arch сначала вызвал непонимающие взгляды, но после некоторых усилий они просветлели. Дэниел объявил: «Raspbian рулит!», а вот Ханне понравилась Risc OS, и с виду, и в плане работы: «Risc OS куда симпатичнее».

Вот так. Юное поколение сказало свое веское слово!

Вердикт

Raspbian ★★★★★
Risc OS ★★★★★
Arch ★★★☆☆
Android ★★☆☆☆
Plan 9 ★★☆☆☆

» Детям понравился Raspbian, и Risc OS, и большой пакет леденцов!

Воспроизведение мультимедиа

Мультимедиа-центрик – дело хорошее, но сразу ли он запоет?

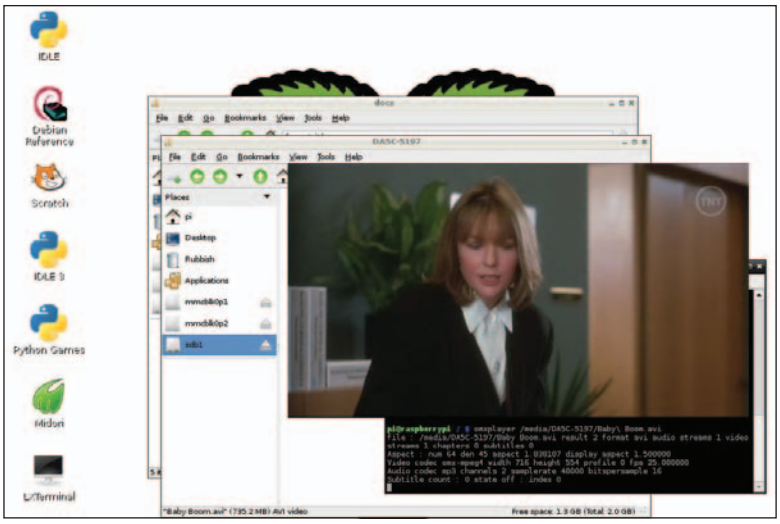
Воспроизведение мультимедиа – тема споров в Linux. Одни дистрибутивы предоставляют свежие кодеки и программы, другие – нет. Прелесть в том, что вы свободны скачать и установить приглянувшийся плейер и настроить его на свой вкус. Но как выбранные нами ОС справятся с задачей немедленного воспроизведения мультимедиа?

Android нас удивил, прилично воспроизведя наш фильм HD-качества. Увы, на 256-МБ RPi воспроизведение через две минуты застопорилось и отказалось двигаться дальше. Как ни странно, в этом пункте нам пришлось заново писать образ на SD-карту, поскольку Android отказался грузиться. 512-МБ версия показала немного лучший результат: видео остановилось через 15 минут. Видео было вполне различимым, хотя во время поворотов и перемещений камеры все сливалось; но мы решили, что это было вызвано ограничениями возможностей ОС по умолчанию. Воспроизведение MP3 было неплохим – на уровне воспроизведения через аудиопорт; однако 256-МБ модель практически не удалось превратить в достойное оборудование для воспроизведения мультимедиа.

Raspbian по умолчанию не предлагал для воспроизведения фильмов или MP3 вообще ничего, и это неприятно, поскольку включение *OMXplayer* по части размера отнюдь его не надсадило бы. Мы понимаем, что одна из целей Raspberry Pi – помочь вылазке пользователей в мир открытого кода, Linux и т. д., и, учась использовать командную строку и репозитории программ, пользователь начинает лучше понимать среду, в которой работает.

Risc OS оказалась немногим лучше – и фильмы, и MP3 отказались грузиться, как и в Plan 9. Хотя и Risc OS, и Plan 9 предоставляют неплохие возможности для просмотра мультимедиа, программы по умолчанию не в состоянии открыть даже самые популярные медиа-файлы – но если повозиться с настройкой, кое-что скачать и установить, то вполне смогут.

Arch точно такой же. Он загружается в Terminal, так что на наличие хоть чего-то графического, способного немедленно приняться за работу, надежды мало. Arch можно превратить в превосходное мультимедиа-устройство, но сложность его освоения может отпугнуть неопытных пользователей.



› Хотя Raspbian ничего не предлагает по части фильмов и MP3, в долгосрочной перспективе у него лучший потенциал для воспроизведения мультимедиа.

Вердикт

Raspbian ★★★★☆
Risc OS ★★★☆☆
Plan 9 ★★★☆☆
Android ★★☆☆☆
Arch ★★☆☆☆

» Android работает на модели 512 МБ, но у Raspbian перспективы лучше.

Подход к программам

Пусть берут наши ОС, но им не отнять у нас свободы!

Все тестируемые здесь операционные системы предоставляют свой исходный код либо на домашних страницах ОС, либо на других соответствующих ресурсах; поэтому по сути своей они свободны и, скорее всего, таковыми и останутся, если, конечно, не изменится ситуация; но в этом случае сообщество пойдет в штыковую за считанные секунды.

Код Risc OS для Raspberry Pi открыт, это версия 5.19, поддерживаемая группой добровольцев; однако последняя версия Risc OS 6 требует, чтобы пользователь подписался на Select scheme за £99 в год, и является полностью проприетарной программой, не предоставляющей доступа к исходному коду.

Plan 9, Arch, Android и Raspbian, как нам всем известно, свободны для любых действий, установки и разработки, в соответствии с моралью как Linux в целом, так и Raspberry Pi Foundation.

Общее отношение к свободе ПО со стороны коллективных проектов одинаково; расширение, дальнейшее обучение и использование каждого из проектов ведется в соответствии с различными идеями и правилами сообщества открытого кода и сообществ, управляющих проектами.

С этим тестом справиться довольно непросто – в вердикте каждая получит пять звезд, потому что каждая свободна, и их подход – это подход свободного кода. Было бы несправедливо признать Risc OS не относящейся к открытому коду из-за закрытого характера Risc OS 6, поскольку сообщество, стоящее за открытой 5.19 для RPi, отличается большим энтузиазмом и более чем способно управлять свободным ПО.

› Risc OS 6 требует, чтобы пользователь подписался на Select за £99 в год.

Вердикт

Android ★★★★★
Arch ★★★★★
Plan 9 ★★★★★
Raspbian ★★★★★
Risc OS ★★★★★

» Пусть это воспримут как отмазку, но пять звезд каждой за свободу! Да!

Операционки для Raspberry Pi

Вердикт

› Победитель — Raspbian, но остальные участники быстро его нагоняют!

Из всех операционных систем, которые мы здесь рассмотрели, в лидеры выбивается Raspbian, будучи самой удобной в использовании, самой привлекательной и обладающей самым широким выбором программ по умолчанию, и т.д.; однако другие конкурсантки дышат ОС от Foundation в затылок.

Для новичка Raspbian воистину станет стартовой площадкой в его путешествии не только в чудесный мир самого Raspberry Pi, но и в огромный мир Linux и альтернативных ОС. Однако как только новичок освоится с тонкостями операционной системы и работы RPi, весьма вероятно, что в качестве основной ОС для Raspberry Pi он установит какой-нибудь Arch, Risc OS или Plan 9.

Android для RPi – определенно интересный проект, и он, без сомнения, еще станет одним из основных дистрибутивов для RPi, но пока что над ним стоит поработать, и, если честно, Raspberry Pi даже близко не подходит к мощи устройств последнего поколения на Android.

Энергию — людям

Энергопотребление здесь является ограничивающим фактором, и здесь следует воздать по заслугам Arch и Plan 9. Эти минималистские и простые системы позволят RPi просто летать, даже не приближаясь к пределу своих ресурсов, а вот Raspbian без должной настройки может оказаться раздутым монстром.

Если бы Foundation произвел устройство помощнее, количество операционных систем для него сразу бы утроилось. Однако проблема здесь кроется в стоимости устройства. Сейчас цена £25 является одним из самых привлекательных моментов в проекте RPi; а чем больше наворотов, тем выше стоимость, так что на данном этапе нас вполне все устраивает, в том числе и имеющиеся операционные системы.

Как и предполагает рубрика, конкуренции здесь нет; есть просто возможность для пользователя попробовать свой Raspberry Pi, установить ОС, приспособить ее по велению души, научиться чему-то, и – как и положено в Linux – начать использовать созвучным себе образом.

Поэтому хотя Raspbian и набирает высший балл, все же не забывайте и о других проектах операционных систем, которые стоят поодаль, ожидая, что их заметят и начнут с ними эксперименты.

> «Если что и превращает Linux в народную ОС, то это сообщество.»

I Raspbian ★★★★★

Сайт: goo.gl/4WwPJ

» Raspbian превосходен для начинающих с Pi. Ветераны, обратите взоры вдаль.

II Risc OS ★★★★☆

Сайт: goo.gl/pn2NP

» Немного усилий — и Risc OS станет полнофункциональной ОС рабочего стола.

III Arch Linux ARM ★★★☆☆

Сайт: goo.gl/L5HtQ

» Arch требует большего опыта, но в конечном итоге более мощный.

IV Android for RPi ★★☆☆☆

Сайт: goo.gl/w4eoy

» Едва тянет на 256-МБ RPi, но со временем ошибки будут устранены.

V Plan 9 ★★☆☆☆

Сайт: goo.gl/qtN7E

» Выглядит старомодно, но, имея терпение, его можно сделать таким, как надо.

Обратная связь

Вы согласны с нашей оценкой ОС для Raspberry Pi? Поделитесь с нами мнениями по адресу lxf.letters@futurenet.co.uk.

Рассмотрите также...

Если эти операционные системы не вызвали у вас особого интереса, рассмотрите PiBang, дистрибутив Linux, вдохновленный популярным Crunchbang Linux. Благодаря превосходным Openbox, Tint2 и Nitrogen, PiBang не только отлично выглядит, но и предоставляет быструю, подвижную и стабильную среду.

PiBang включает *OMXplayer* и *VLC*, но, как указано на сайте PiBang, «*VLC* на данный момент не воспроизводит видео на Raspberry Pi на нормальном уровне, он установлен для тестирования и для воспроизведения аудио». PiBang предлагает полноценную среду рабочего стола, поэтому из него были удалены обучающие программы ради *AbiWord*, *Conky*, *GIMP* и т.п. Он использует те же источники программ, что и Raspbian, что добавляет совместимости. Фактически, включи мы PiBang в число наших тестируемых ОС, велик шанс, что он бы стал победителем. Почему же мы его не включили? Ну, это ОС, которая ничем особым не выделяется. Хотя мы не уверены в этом: она очень, очень хороша. LXF

Linux — что это?

Нейл Ботвик копается в лучшей в мире операционной системе, выясняя, что же заставляет ее работать.

Слово «Linux» — одно из самых часто встречающихся в этом журнале, но что оно означает? Оно означает разное для разных людей — от пуристов, для которых это в первую очередь ядро, до сторонников GNU, которые считают его частью GNU/Linux, и новых пользователей, полагающих, что это второе название Ubuntu. На самом деле Linux — все это сразу, а выбор зависит от вашей точки зрения. Строго говоря, термин "Linux", используемый отдельно, относится к ядру операционной системы, а GNU/Linux — это вся операционная система, объединяющая ядро Linux и инструменты GNU; одно будет совершенно бесполезно без другого (или какой-либо его альтернативы).

Если добавить сюда еще и приложения, и некоторые инструменты, позволяющие управлять всем этим в целом, вы получите дистрибутив, например, Ubuntu.

За долгие годы мы рассказали практически обо всех отдельно взятых компонентах, составляющих операционную систему, известную нам как Linux, но они не работают по отдельности – они должны объединяться вместе, и здесь мы попытаемся объяснить, как из частей складывается целое и что эти части делают.

Что такое ОС? Что такое дистрибутив?

Операционную систему можно определить как программу, необходимую для того, чтобы приложения работали на оборудовании – как таковая, она состоит из нескольких переплетенных уровней. В самом сердце ее лежит ядро, которое взаимодействует с оборудованием через драйверы и позволяет другим программам использовать это оборудование. Поверх него располагаются различные слои, работающие с устройствами ввода, сетевыми устройствами, устройствами звука и видео. Обычно вам незачем об этом знать. Знания, конечно, бывают полезны при возникновении каких-либо проблем, но даже тогда они не так уж важны, особенно если ваш компьютер есть кому отремонтировать. Однако, раз уж вы читаете этот журнал, есть вероятность, что вас интересует происходящее «внутри» Linux, и мы попытаемся дать вам представление об этом – и о том, кто за что отвечает.

Дистрибутив Linux – это способ распространения основанной на Linux операционной системы и сопутствующих программ. В самом начале это были просто файлы, необходимые для ОС, плюс способ их установки на компьютер. Постепенно дистрибутиву начали требоваться менеджеры пакетов, инструменты обновления, настройка GUI и множество иных удобств. Тем не менее, при появившихся дружелюбии к пользователю и внешнем лоске (или не появившихся, если ваш дистрибутив – Gentoo) все дистрибутивы по-прежнему представляют собой Linux.

Ядро

Нервный центр в сердцевине вашей операционной системы Linux.

Ядро — это бьющееся сердце системы, но что оно такое? Ядро — это программный интерфейс к оборудованию компьютера. Оно общается с CPU, памятью и прочими устройствами от имени любой программы, установленной на компьютере. И как таковое, является самым низкоуровневым компонентом среди программ — и самым важным. Если в ядре есть проблема, эта проблема отражается на любой программе, работающей на компьютере.

Ядро Linux монолитно, т. е. все основные сервисы ОС работают в нем. Альтернативный подход – микроядро, где большая часть работы делается внешними процессами, а ядро занимается всего лишь координированием.

Чисто монолитное ядро неплохо работало раньше, когда пользователи компилировали ядро под свое оборудование; но сейчас такое количество комбинаций оборудования, что встраивание их всех в ядро приведет к появлению неподъемного файла. Поэтому ядро Linux стало модульным: ключевые функции находятся в файле ядра (его можно увидеть в **/boot** как **vmlinuz-version**), а опциональные драйверы встроены в виде отдельных модулей в **/lib/modules** (файлы **.ko** в этой директории).

Например, 64-битное ядро Ubuntu 12.10 имеет размер 5 МБ, и есть еще 3700 модулей, занимающих более 100 МБ. На конкретной машине нужна только часть из них, и было бы сущим безумием пытаться загрузить их все с основным ядром. Вместо этого ядро определяет используемое оборудование и загружает соответствующие модули, которые становятся частью ядра в памяти, так что в загруженном виде оно по-прежнему остается монолитным, даже когда распределяется по тысячам файлов. Это позволяет системе реагировать на изменение оборудования. Если подключить устройство USB, загрузятся модуль **usb-storage** и модуль файловой системы, необходимый для того, чтобы его подмонтировать. Вот почему при добавлении оборудования так редко приходится устанавливать новые драйверы: все они уже есть, и просто ждут, когда вы купите и подключите новые игрушки. Компьютеры, работающие на специализированном оборудовании, которое не изменяется, например, серверы, обычно имеют ядро с уже скомпилированными необходимыми драйверами и отключенной загрузкой модулей, что добавляет небольшую долю безопасности.

Если вы компилируете свое ядро, хорошо бы взять за правило встраивать в него драйверы для тех устройств, которые всегда используются, например, для сетевого интерфейса и файловых систем жесткого диска, а для всего остального создать модули.

И еще модули

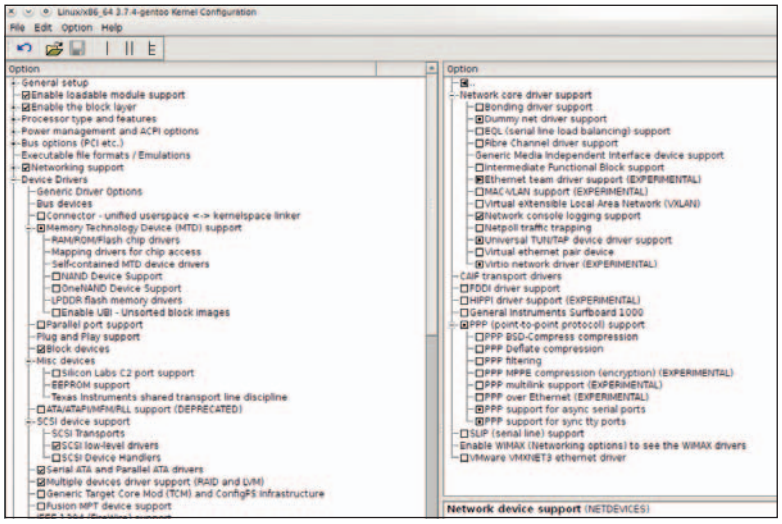
Огромное количество модулей, большинство из которых – драйверы устройств, стало одной из сильных сторон Linux: по умолчанию поддержано столько устройств, что драйверы качать и устанавливать откуда-то еще совершенно незачем. Но некоторые устройства пока не охвачены встроенными в ядро модулями, обычно по причине того, что код еще очень нов или его лицензия не позволяет включать его в ядро (да, ZFS, это про тебя).

Самый известный пример – драйверы для карт Nvidia. Обычно именуемые сторонними модулями, хотя Ubuntu также использует термин «ограниченные драйверы», они устанавливаются из вашего менеджера пакетов, если их поддерживает ваш дистрибутив. В ином случае их придется компилировать из исходника, что нужно будет проделывать регулярно, при каждом обновлении ядра, поскольку они привязаны к ядру, для которого компилировались.

> «Поддержано столько устройств, что драйверы качать незачем.»

Предпринимались некие попытки автоматизировать этот процесс, например, DKMS (Dynamic Kernel Module Support – динамическая поддержка модулей ядра), когда при установке ядра автоматически перекомпилируются все сторонние модули, делая процесс обновления ядра почти таким же безболезненным, как процесс обновления пользовательских приложений.

Говоря о ядре, обязательно жонглируют такими понятиями, как «область ядра» и «пользовательская область». Область ядра – это память, доступ к которой есть только у ядра; ни одна пользовательская программа (то есть нечто помимо ядра и его модулей) не может ничего здесь записывать, поэтому сторонняя программа не может повредить работе ядра. А пользовательская область – это область, доступ к которой может получить любая программа с соответствующими привилегиями. Это способствует стабильности и безопасности Linux, поскольку ни одна программа, даже работающая от имени суперпользователя-root, не может напрямую разрушить ядро.



› Количество опций при компиляции ядра реально устрашает, и большинство из них касается поддержки устройств. Разве вы не рады, что производители дистрибутивов делают все это вместо нас?

Порядок загрузки

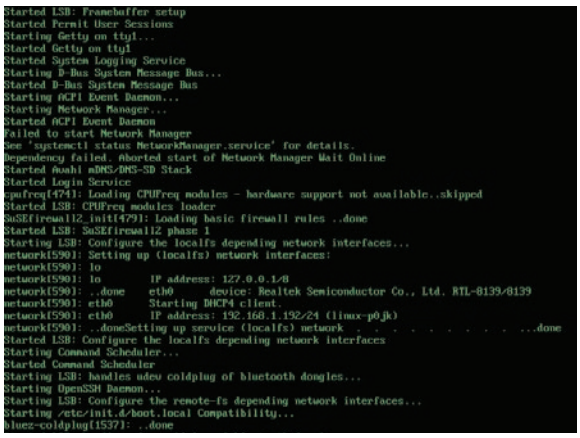
Таинственное чередование вспышек и звуков при запуске.

Большинство дистрибутивов при загрузке сразу прикрываются заставкой, и мы не можем наблюдать за происходящим. А происходит-то много чего — как до появления заставки, так и после. Первым загружается BIOS с материнской платы. Она ищет устройство загрузки и загружает с него код.

Если жесткий диск использует традиционную систему деления на разделы DOS, она содержится в Master Boot Record (MBR) диска – всего 512 байт. 64 байта используются для хранения таблицы разделов для диска (вот почему доступны только четыре первичных раздела), а остальные 446 байт остаются для кода программы-загрузчика (обычно это *Grub*). В 446 байт особо много функций не впихнуть, поэтому все, что делает этот код – загружает остаток загрузочного кода откуда-то еще с диска, из того места, которое устанавливается при задании кода MBR загрузчиком.

> «Файл конфигурации выдает загрузчику адрес ядра Linux.»

В поисках списка опций загрузки загрузчик считывает свой файл конфигурации – обычно это **/boot/grub2/grub.cfg** – и либо отображает меню, либо переходит прямиком к загрузке по умолчанию. На данной стадии Linux никак не задействован: работает только код загрузчика. Файл конфигурации выдает загрузчику адрес ядра Linux и всех необходимых файлов initramfs, а также любые другие настройки, такие, как раздел root и то, надо ли скрывать это все за заставкой. Для желающих видеть, что происходит начиная с этого момента и дальше, в большинстве дистрибутивов заставку можно убрать, нажав кнопку E для редактирования пунктов меню *Grub*, удалив опции заставки и заглушения звуков и нажав F10 для продолжения загрузки.



Убрав заставку, вы увидите процесс загрузки во всей красе, в том числе и состояние запускаемых процессов.

Зачем нужен ramdisk?

Большинство дистрибутивов используют файл initramfs – главным образом, по той причине, что определенные драйверы, в частности, нужные для загрузки диска (например, контроллеры SATA и код файловой системы), надо загружать вместе с ядром. Для обычного дистрибутива встраивание всех возможных опций в ядро раздует его настолько, что оно окажется неработоспособным, поэтому все встраивается в виде модулей, и те, что нужны для загрузки, включаются в initramfs. Это тип ramdisk'а, загружаемого программой загрузки вместе с ядром (используя процедуру BIOS, чтобы считать его с диска), содержащий все файлы, необходимые для того, чтобы подмонтировать раздел root. Здесь помогает функция распознавания ядром устройств, решая, какие модули загружать, и затем управление передается жесткому диску. Также initramfs используется для вывода на экран заставок, поэтому они и появляются в самом начале процесса загрузки.

После монтирования раздела root, напрямую или через initramfs, всерьез запускается последовательность инициализации. Традиционно она начинается с запуска **/sbin/init**, которая затем запускает все остальное, под контролем **/etc/inittab**, и отвечает за список сообщений при запуске сервисов, которые вы видите на консоли, если у вас нет заставки. Также это позволяет увидеть, в каком месте процесс загрузки подвис или занял больше положенного времени, если у вас возникли такие проблемы.

Новые опции

Время идет, и все эти системы подвергаются изменениям. На самом последнем оборудовании BIOS заменяется на UEFI, хотя если вы установили программу загрузки, разницы вы не почувствуете.

GNU в составе GNU/Linux

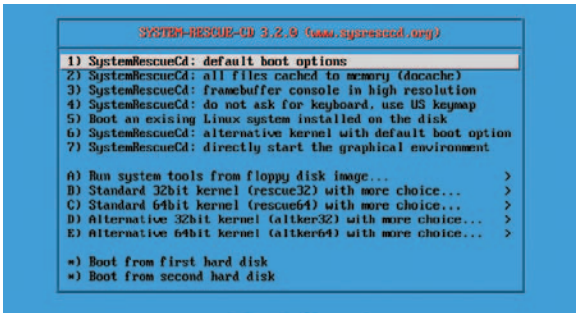
Проект GNU (Gnu's not Unix, Gnu – это не Unix) предвосхитил Linux на несколько лет. Он создал большинство инструментов, нужных компьютеру в начале 1980-х – компиляторы, текстовые редакторы, команды для работы с файлами и директориями и многое другое – но у него не было удобного в использовании ядра (некоторые считают, что их ядро, GNU Hurd, удобным так и не стало). Когда Линус Торвальдс начал работать над своим маленьким проектом в 1991 году, у него было ядро, но не было инструментов для работы на нем. Тогда эти проекты объединились, и родилась GNU/Linux – ОС, которая использует ядро Linux и набор инструментов GNU.

Из GNU пришли не только программы в **/bin** и **/usr/bin**; *glibc* – основная библиотека C, используемая в Linux – это тоже наследие GNU. И при каждом вашем действии на своем компьютере, при каждом вводе команды или щелчке по значку на каком-то уровне запускается программа GNU. Неудивительно, что стойкие приверженцы GNU огорчаются, когда мы именуем нашу ОС Linux, а не GNU/Linux. Но стоит отметить, что никто не отрицает важности аспекта GNU; то, что ОС называют просто Linux вместо GNU/Linux, в большей степени объясняется ленью, нежели политическими причинами – полное название слишком уж громоздкое.

Прости, Ричард, но мы не будем все время говорить «GNU слэш».

Есть также попытки заменить традиционную систему *SysVinit*, которая существует уже много лет. Ubuntu ввел *Upstart*, а Red Hat и Fedora предпочитают *systemd*.

Все они делают примерно одно и то же – запускают задачи, необходимые, чтобы заработала ОС. Отличаются только методы. Основное различие в том, что *SysVinit* последовательна: каждый следующий сервис дожидается завершения запуска предыдущего. Медленно запускаемый сервис тормозит все остальные. *Upstart* и *systemd* запускают сервисы параллельно, избегая подобных узких мест. Конечно, есть те, кто считает, что Linux настолько стабилен, что время загрузки не столь важно – если вы вместо выключения машины можете уйти в спящий режим, перезагрузка становится весьма редким случаем.



› ***Grub* — самая популярная программа загрузки. Но Live CD и устройства USB будут скорее использовать *isolinux* или *syslinux*.**

Библиотеки

Логика разделения функций между программами.

Linux использует библиотеки, чтобы разделить код между приложениями. Если некая программа *foo* использует функции, полезные для кого-то еще, она помещает их в библиотеку — назовем ее *libfoo*. И если другой программе, назовем ее хотя бы *bar*, понадобится та же функция, она всего лишь обратится к библиотеке *libfoo*, не изобретая велосипед заново.

Это означает, что на вашем компьютере имеется всего одна копия кода функции; если какой-то проект обнаружит в ней ошибку, та будет исправлена везде. Здесь также возникает понятие зависимостей: и *foo*, и *bar* зависят от *libfoo* и бесполезны без нее. Это приводит к феномену «ада зависимостей», когда попытка установить программу завершается ошибкой и списком недостающих зависимостей, а попытка установить их притягивает еще больше зависимостей. В наши дни это скорее неприятное воспоминание, поскольку репозитории дистрибутивов стали более всесторонними, а менеджеры пакетов лучше справляются с сортировкой.

> «Команда ldd покажет все библиотеки, нужные программе.»

Если вы работаете со своим менеджером пакетов и репозиториями, то обо всех зависимостях позаботятся, и вам даже не придется о них думать. Попробуйте установить пакет **somerandom.deb** или **somerandom.rpm**, скачанный с некоего сайта – назовем его www.somerandomsite.com; и вы вмиг поймете, почему стоит все перевалить на менеджер пакетов. Одно из предлагаемых решений – статическая компиляция программ. Это означает, что вместо динамических ссылок на код в *libfoo* и подгрузке при необходимости во время работы, и *foo* и *bar* включали бы код в свои исполняемые программы. Тогда каждый файл программы станет отдельным объектом, без всяких зависимостей; но он будет намного объемнее, и если в коде *libfoo* обнаружится ошибка или брешь в системе безопасности, и *foo*, и *bar* придется перекомпилировать и заново включать в пакеты вашего дистрибутива, чтобы исправить ситуацию. На невстраиваемых устройствах предпочитают применять динамические ссылки, а вот в initramfs, вызываемом во время загрузки, удобны статические: тогда не надо включать библиотеки в образ ramdisk. Любопытствующие могут узнать, на какие библиотеки ссылается программа – командой **ldd**.

```
ldd /usr/bin/someprogram
```

покажет все библиотеки, нужные программе, и библиотеки, нужные этим библиотекам, и т.д., пока вы не доберетесь до *libc* – прадедушки библиотек Linux.

Менеджеры пакетов

Замечательная гибкость дистрибутивов Linux означает, что большинство элементов можно изменить. Приложения по умолчанию, рабочие столы, даже ядро можно поменять, так что лучше всего считать дистрибутив Linux – какой-нибудь Fedora или Ubuntu – всего лишь стартовой точкой для индивидуальной настройки.

Но есть одна вещь, не так легко поддающаяся изменениям, а именно – менеджер пакетов: единственный способ его сменить – это взять другой дистрибутив. Попробуйте, например, сравнить *Yast* от SUSE с *Synaptic* в Debian, и вы удивитесь, до какой степени столь важный инструмент может изменить ваш опыт работы в Linux.

```
[nelz@hactar ~ 0]% ldd /usr/bin/k3b
        linux-vdso.so.1 (0x00007fffe61ff000)
        libk3bdevice.so.6 => /usr/lib64/libk3bdevice.so.6 (0x00007ffd9cbec000)
        libk3blib.so.6 => /usr/lib64/libk3blib.so.6 (0x00007ffd9c811000)
        libkcddb.so.4 => /usr/lib64/libkcddb.so.4 (0x00007ffd9c5cc000)
        libkfile.so.4 => /usr/lib64/libkfile.so.4 (0x00007ffd9c31b000)
        libkio.so.5 => /usr/lib64/libkio.so.5 (0x00007ffd9be4b000)
        libknotifyconfig.so.4 => /usr/lib64/libknotifyconfig.so.4 (0x00007ffd9bc370
        libkde3support.so.4 => /usr/lib64/libkde3support.so.4 (0x00007ffd9b932000)
        libsolid.so.4 => /usr/lib64/libsolid.so.4 (0x00007ffd9b636000)
        libQtWebKit.so.4 => /usr/lib64/qt4/libQtWebKit.so.4 (0x00007ffd99b7c000)
        libkcmutils.so.4 => /usr/lib64/libkcmutils.so.4 (0x00007ffd99938000)
        libQt3Support.so.4 => /usr/lib64/qt4/libQt3Support.so.4 (0x00007ffd99455000
        libQtXml.so.4 => /usr/lib64/qt4/libQtXml.so.4 (0x00007ffd99210000)
        libkdeui.so.5 => /usr/lib64/libkdeui.so.5 (0x00007ffd98b73000)
        libkdecore.so.5 => /usr/lib64/libkdecore.so.5 (0x00007ffd98682000)
        libQtCore.so.4 => /usr/lib64/qt4/libQtCore.so.4 (0x00007ffd981a4000)
        libQtGui.so.4 => /usr/lib64/qt4/libQtGui.so.4 (0x00007ffd974f3000)
        libstdc++.so.6 => /usr/lib/gcc/x86_64-pc-linux-gnu/4.6.3/libstdc++.so.6 (0x
        libc.so.6 => /lib64/libc.so.6 (0x00007ffd96e45000)
        libpthread.so.0 => /lib64/libpthread.so.0 (0x00007ffd96c28000)
        libm.so.6 => /lib64/libm.so.6 (0x00007ffd96936000)
        libsamplerate.so.0 => /usr/lib64/libsamplerate.so.0 (0x00007ffd965ca000)
        libdl.so.2 => /lib64/libdl.so.2 (0x00007ffd963c6000)
        libdvdread.so.4 => /usr/lib64/libdvdread.so.4 (0x00007ffd961a8000)
        libQtNetwork.so.4 => /usr/lib64/qt4/libQtNetwork.so.4 (0x00007ffd95e5e000)
        libXrender.so.1 => /usr/lib64/libXrender.so.1 (0x00007ffd95c53000)
```

› **Общие библиотеки делают систему более эффективной, разделяя код между приложениями. Вот всего лишь часть библиотек, на которые ссылается программа прожига дисков *K3b*.**

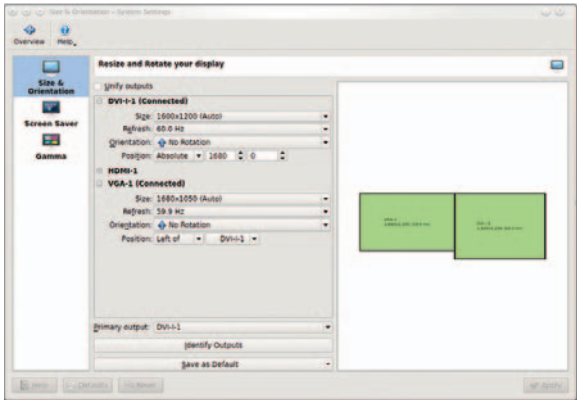
Графика

Как вашему Linux удается столь классно выглядеть.

Стандартной основой для графического интерфейса является *X Window System*. Всякие KDE и Gnome обеспечивают интерфейс пользователя и внешнюю привлекательность, но именно через *X* они общаются с оборудованием. Долгие годы *X* требовал массу мистических опций настройки и требовал наличия длинного файла конфигурации, содержащего такие вещи, как моделайны, задающие частоту пикселизации и синхронизации.

Ныне большинство систем обходятся без всяких файлов настройки. Солидные улучшения в распознавании оборудования означают, что система с одной видеокартой и одним дисплеем будет «просто работать». Возможно, понадобятся дополнительные драйверы для 3D-ускорения, если вы, например, используете карту Nvidia, но в ином случае вы просто загружаете компьютер и принимаетесь щелкать мышью. *X* построен на архитектуре клиент/сервер: сам по себе он работает как сервер, поддерживая отображение; затем клиентская программа общается с сервером, сообщая ему, что и где изобразить.

> «Ныне большинство систем обходятся без файлов настройки.»



› Инструменты KDE вроде настройщика монитора помогают в таких ситуациях, как подключение второго монитора, но для единственного дисплея *X* вообще не надо настраивать.

Унаследованные функции

Возможно, это покажется чрезмерным усложнением, но он использует локальные сокеты для общения клиента с сервером, так что производительности удара не наносит. Явное преимущество этого метода – то, что клиенту и серверу не обязательно находиться на одном и том же компьютере. Вы можете соединиться с другим компьютером через SSH, и при условии, что настройка это позволяет, запустить программу на удаленном компьютере и заставить его GUI отображаться на локальном экране. Это отличается от VNC, потому что только окно одного приложения может отображаться локально, и оно появляется только локально – не на удаленном компьютере. Соединение VNC зеркально отображает весь рабочий стол на обоих компьютерах, поэтому предпринимаются шаги для разработки более простых методов работы с графическим дисплеем. Самый продвинутый – *Wayland*. *Wayland* использует совершенно другой подход; в нем не только нет старой настройки клиент/сервер, но к тому же рендеринг окон и других элементов дисплея передан клиентским приложениям, обычно *OpenGL* или *Cairo*. Это упрощает *Wayland*; *X* содержит большое количество наследуемого кода рендеринга, который требуется спецификациями *X*, но при этом никогда не используется. Передавая управление клиентам, *Wayland* может стать более легким, более эффективным и защищенным. Это также означает, что графические программы, с которыми вы работаете, имеют больше контроля над способом отображения GUI.

Скромные труженики — демоны

Отключив заставку, которую большинство дистрибутивов используют для сокрытия сообщений о загрузке, вы увидите экран, заполненный сообщениями о запускаемых сервисах. Что это за сервисы, и все ли они нужны? Сервисы – это программы, работающие в фоновом режиме; они-то и делают компьютер работоспособным. Одни работают с сетью, другие – с определением и настройкой оборудования, но большинство – это традиционные программные сервисы, или демоны, которые при необходимости предоставляют функции другим программам. Ответом на вторую часть вопроса, скорее всего, будет «нет». Некоторые из этих сервисов используются почти всеми системами, например, демон *syslog*, который пишет информацию в системные журналы; но другие могут быть не нужны совсем. Зачем вам запускать CUPS, систему печати, если у вас нет принтера? Аналогично, может быть не нужен сервер базы данных *MySQL*, или демон SSH, если у вас всего один компьютер в сети. Так что, потратив на эксперименты где-то полчаса, вы сможете выкроить пару секунд времени загрузки.

Можно сэкономить ресурсы, не запуская ненужные сервисы; однако, будучи однажды загружены, эти демоны почти не потребляют системных ресурсов, и если их не вызывать, то даже требуемая им память отдается другим программам,. Так что просто отключите сервисы, которые, по вашему мнению, вам никогда не понадобятся. Если они терпеливо прислушиваются к сетевому порту или сокету, работа ваших клиентских программ станет чуть эффективнее. Программам нет нужды включать или загружать код для открытия или закрытия лог-файлов, или записи в них – они просто вызывают функцию **syslog()** с текстом для журнала, а об остальном заботится демон. *Syslog* является важным сервисом – если появились проблемы, именно сюда стоит заглянуть в первую очередь, поскольку большинство программ отправляют сообщения об ошибках в системный журнал (обычно в **/var/log/messages**).

› Можно сократить время загрузки, запуская только те сервисы, которые вам нужны.

Почему фоновые программы назвали демонами? Объяснений несколько – я предпочитаю такое: в греческой мифологии так звались создания, решавшие задачи, с которыми не хотели возиться боги.

Сеть

Как ваш компьютер общается с другими.

Работа в сети крайне важна для Linux. Даже на отдельной машине без соединения с локальной сетью, не говоря уж о соединении с Интернет, сетевая работа все же используется. Многие сервисы работают по модели клиент/сервер, где сервер работает в фоне, ожидая инструкций от других программ. Даже такая важная штука, как системный регистратор (system logger), работает в качестве сетевого сервиса, разрешая другим программам писать в лог-файлы. Графическая система *X* тоже является сетевой, где *X*-сервер работает с рабочим столом и программами, сообщая, что им надо отобразить. Вот почему так просто запустить *X*-программы на удаленном рабочем столе — с точки зрения работы системы это не очень отличается от локального открытия окна.

Запуск **ifconfig** всегда будет отображать по крайней мере один интерфейс под названием lo, с адресом 127.0.0.1 – он используется компьютером для общения с самим собой, что для компьютеров вполне нормально (в отличие от людей). Остальная, и большая, доля сетевой работы основана на TCP/IP, либо в проводном Ethernet, либо в беспроводном соединении; но используются и другие типы сетей. Все дистрибутивы и рабочие столы включают хорошие инструменты для настройки и работы в сетях TCP/IP, от вездесущего *NetworkManager* до индивидуальных инструментов, таких, как инструмент настройки сети Gnome или *Yast* от OpenSUSE.

Среди недавних дополнений к сетевой работе – технологии 3G и PAN (Personal Area Network), такие, как Bluetooth. Использовать 3G-модем обычно просто, или через NetworkManager, или через средства настройки PPP-соединения вашего рабочего стола. Да, 3G-модемы и правда работают, как модемы, со скриптами набора номера и прочим, но без шипения и попискивания, обычно сопровождающих соединение (юным читателям последнее предложение ничего не скажет). Большая часть проблем с 3G возникают из-за попыток настроить их в области со слабым сигналом, а не из-за поддержки оборудования или ПО в Linux, как я недавно вновь убедился.

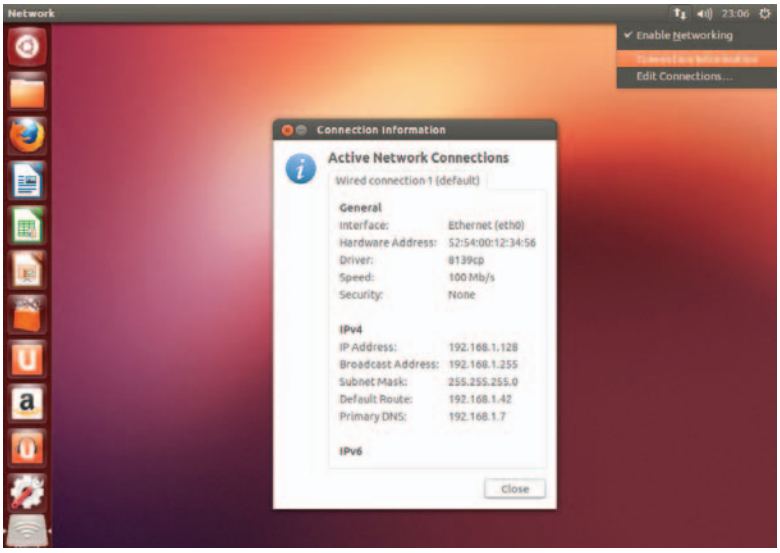
Королевский протокол

Все большую важность приобретает Bluetooth, по мере распространения мобильных устройств и роста числа устройств ввода и вывода, применяющих его. Выигрывают от этого не только хозяева телефонов и планшетников: мышь и наушники Bluetooth расширят возможности ноутбука, когда вы сидите за столом, не заставляя ничего подключать к нему перед началом работы. PulseAudio (см. раздел про звук) делает это еще проще, поскольку умеет переключаться между распознанными устройствами.

Хранение

Хранение данных на жестком диске может состоять из нескольких уровней. Все физическое хранение (в отличие от сетевого хранения) в Linux вращается вокруг блочных устройств, именуемых так потому, что диски хранят данные в блоках. Блочные устройства типа **/dev/sda1** действительно имеют дело с блоками, физическими областями на диске, и их скоплениями в качестве раздела диска. Поверх всего этого находится файловая система, то есть способ хранения данных в разумной структуре директорий и файлов, содержащих как данные, так и метаданные.

В чем их разница? Скажем, вы сохраняете файл с текстом. Данные в этом файле – текст, но у файла есть другие атрибуты: владелец, время создания, время последнего изменения, время последнего просмотра и кто имел право изменять или просматривать файл. Эту информацию вы увидите, применив к файлу **ls -l** или проверив его свойства в менеджере файлов, и она сохраняется файловой системой. Стандартная используемая сегодня файловая система – ext4, но есть и альтернативы: ext3, ReiserFS, XFS, JFS и, конечно, FAT и NTFS из мира Windows.



› Сетевая организация — ключ к работе системы Linux. Интерфейс localhost настраивается автоматически; а для всего остального у нас есть программы вроде *NetworkManager*.

Другие Linux'ы

Все, о чем мы говорили, относится к Linux, работающему на настольном компьютере, ноутбуке или сервере – традиционном компьютерном оборудовании, если угодно, но есть и иные среды, где применяется Linux. Многие встраиваемые устройства, от роутеров до PVR и приставок, работают на Linux, и во многом он схож с Linux, который мы знаем и любим. Если ваш роутер разрешает доступ SSH, вы будете чувствовать себя, как дома, как только войдете.

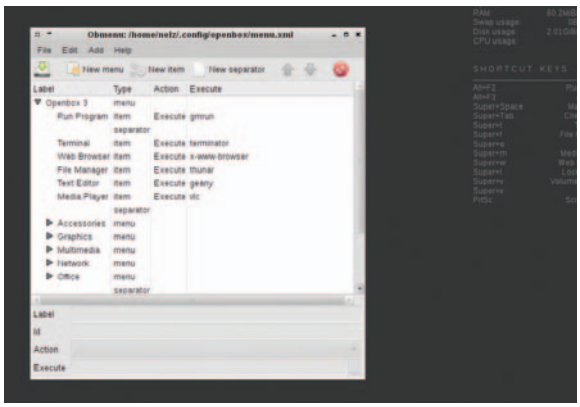
Есть еще один класс устройств, который в последние годы обрел огромную популярность, и на котором работает совсем другой Linux. Нетрудно догадаться, что речь идет о смартфонах и их планшетных родичах, работающих на Android. Android – тоже Linux, он использует ядро Linux, но это не GNU/Linux. Ядро может быть основано на том же самом исходном коде, но все, что работает на нем, совершенно другое. Принципы в некотором смысле схожи, но реализация их иная – хотя, попав в оболочку на своем телефоне, вы увидите знакомые инструменты командной строки.

Рабочие столы

Gnome, KDE, Cinnamon, Unity – мы зовем их интерфейсом пользователя.

Если самым нижнем уровнем системы считать ядро, то самый высший — это интерфейс пользователя. Все остальное, от ядра до драйверов и интерфейсов оборудования, бесполезно, если вы не можете использовать компьютер. Обычно это означает графический рабочий стол, и здесь мы сталкиваемся с другими уровнями. *X* (или, возможно, *Wayland* в будущем) просто обеспечивают чистый холст. Вам нужно нечто, предоставляющее все прелести оконного интерфейса, иногда менеджер окон.

Ранее менеджеры окон были отдельными системами, и по-прежнему имеется много всяких *OpenBox* или *Enlightenment*, но сейчас они нередко являются частью более крупной среды рабочего стола. Строго говоря, менеджер окон отвечает за работу с окнами, их открытие, закрытие, перемещение и другие манипуляции. Со временем менеджеры окон росли и обретали дополнительные функции, такие как панели задач и меню запуска программ, пока не превратились в среды рабочего стола.



› Есть множество легковесных менеджеров окон — типа *OpenBox*, работающего на CrunchBang.

Подборки программ

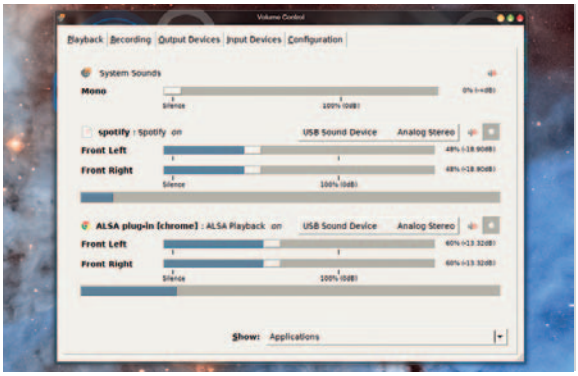
Среда рабочего стола – это просто в большей или меньшей степени интегрированная подборка утилит для обеспечения функций, необходимых для функционирования рабочего стола. Запуск программ, манипуляции с их окнами, отслеживание происходящего и включение программ для общения друг с другом – все это функции рабочего стола, но все же в сердцевине их находится менеджер окон – *KWin* для KDE и *Metacity* в Gnome, если назвать только два. Отличает менеджер окон от рабочего стола уровень интеграции. Это особенно очевидно в KDE, где все работает вокруг общей сердцевины, и программы не только общаются друг с другом, но экземпляр одной программы может быть встроен в окно другой.

И хотя использовать *KWin* в Gnome особого смысла нет, вы, возможно, захотите попробовать один из более узких менеджеров окон, предлагающих больший контроль за работой с окнами или применяющих другие методы их отображения. Существуют менеджеры окон мозаичного типа, например, *awesome* и *xmonad* – они автоматически изменяют размер окон, чтобы все уместилось на экране (в KDE есть собственная такая опция). Есть также менеджеры окон, созданные для управления без мыши, и минимальные менеджеры окон, полезные для специфичных систем, запускающих одну программу в полноэкранном режиме и не одобряющих загромождения пространства виджетами.

› Gnome, KDE, Unity, Cinnamon, Mate — у нас нет недостатка в выборе, когда речь идет о средах рабочего стола, однако многие ли из вас пробовали более двух?

Звук

Некогда мучительная тема.



› Если вам скажут, что PulseAudio сложен, лучше не спорьте. Какое вам до этого дело, если все просто работает.

Звуковая система Linux претерпела множество изменений. Долгие годы у нас была OSS (Open Sound System), до появления ALSA (Advanced Linux Sound Architecture). ALSA по-прежнему с нами, но теперь есть еще PulseAudio и Jack.

ALSA поддерживает несколько звуковых карт (каждая из них может иметь многочисленные входы и выходы) и предоставляет поддержку аппаратного микширования, если оно предусмотрено оборудованием, и программное микширование, если нет. Тогда устраняется потребность в звуковых менеджерах, некогда обеспечивавших в KDE и Gnome возможность воспроизведения звука более чем одним приложением одновременно. ALSA работает на низком уровне, ближе к оборудованию, поэтому дает низкую латентность. Сейчас поддерживается масса устройств, и после установки дистрибутива звук должен заработать при первой же загрузке. ALSA – комбинация кода ядра и приложений пользовательской области. Она также предоставляет API, чтобы другие программы могли напрямую управлять ей; так, в среду рабочего стола включены кнопки управления микшером.

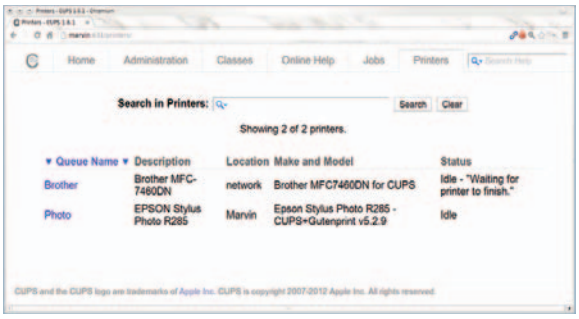
Деятельность PulseAudio

PulseAudio – более новая аудиосреда, но это не замена ALSA. Она располагается поверх аудиосистемы ядра, предоставляя более высокий уровень управления, и работает как сервер, принимая информацию из исходных источников и перенаправляя ее на приемник (устройство вывода результата или программу для записи). Во многих случаях запись осуществляет ALSA, и источником тоже может быть драйвер ALSA – для приложений, не поддерживающих PulseAudio напрямую. Таким образом, приложение может направлять результат на устройство ALSA, которое перехватит поток и отправит его через PulseAudio назад в ALSA. Неудивительно, что PulseAudio считается чрезмерно сложным. Хорошая настройка должна сделать все эти ухищрения невидимыми для пользователя, и именно такова сейчас ситуация с установкой дистрибутивов и PulseAudio, так что в основном ALSA по-прежнему «работает», но с лучшей поддержкой многочисленных устройств. ALSA поддерживает несколько устройств вывода, но по умолчанию у нее глобальные настройки. PulseAudio позволяет направлять музыку через наушники, используя для звонков VOIP наушники Bluetooth, и обеспечивает менее сложное, но столь же полезное разделение – скажем, независимые настройки громкости для приложений. PulseAudio хорошо работает в сети; через него можно поискать другие серверы PulseAudio и воспроизводить аудио через их микрофоны – отлично для потоков музыки в вашем доме.

Для профессиональных аудиоприложений есть звуковой сервер JACK (Jack Audio Connection Kit). Его сильная сторона – обеспечение соединений с низкой латентностью в реальном времени между приложениями для аудио- и MIDI-данных. Для обычного настольного применения он нужен разве что юным музыкантам.

Печать

CUPS и драйверы принтеров.



› Работать с принтерами CUPS не сложнее, чем следовать по ссылкам в браузере, благодаря его встроенному web-интерфейсу.

Хотя открытый код поощряет выбор, и, следовательно, наличие нескольких программ, которые делают одно и то же, но немного по-разному, все же есть области, где вне конкуренции оказывается одна программа. *X.org* используется повсеместно для графического отображения, а в области печати то же положение занимает CUPS.

Если к вашему компьютеру с Linux подсоединен принтер, вам нужны две вещи: CUPS и драйверы для вашего принтера. Во многих случаях они идут вместе. CUPS – это сервер, который работает в фоновом режиме, ожидая запросов на печать. Практически любая программа, которая занимается печатью, распознает Internet Printing Protocol (IPP), на котором говорит CUPS. Нажмите Print в своем текстовом редакторе или браузере, и появится окно, которое покажет ваш принтер, или, если у вас их несколько, предложит выбрать один из них.

Приложению просто нужно отправить данные для печати, обычно в виде PostScript, на CUPS, который и сделает все остальное – даже подождет, пока вы включите принтер. CUPS не нужно даже работать на том же компьютере – это сетевой сервис, а принтер от одного компьютера должен быть доступен всем в той же сети. CUPS бесполезен без драйверов, объясняющих ему, как общаться с принтером. По умолчанию он включает массу драйверов, и еще больше станет доступно при установке пакета драйверов *gutenprint* (настолько, что ваш дистрибутив мог бы и установить этот пакет по умолчанию). HP также предоставляет пакет драйверов (в том числе и для сканера) под названием *hplip*, который вам понадобится, если вы используете их продукцию.

Однако некоторые изготовители принтеров настаивают на том, чтобы предоставлять собственные драйверы, а не включать их в CUPS – обычно это обусловлено лицензионными соображениями. Тогда вам следует порыскать по сайту производителя вашего принтера, найти пакет драйверов, подходящий для вашей системы, и установить его отдельно. После этого драйверы должны появиться в CUPS и в инструменте настройки принтера вашего дистрибутива. Другой вариант – до покупки принтера поискать на linuxprinting.org более просвещенного производителя. LXF

Бархатный диктатор

Django покоряет Интернет. Джейкоб Каплан-Мосс раскрывает LXF планы командования.

За два года со времени нашей последней беседы с одним из основателей Django история его головокружительного взлета получила продолжение. Заключительную часть нашего учебника по Django вы найдете в этом же номере, а Pinterest, Instagram и Mozilla уже вовсю используют его для создания эффективных динамических web-интерфейсов, вдобавок прекрасно масштабируемых. Джейкоб [Jacob Caplan-Moss], пожизненный бархатный содиктатор, стоял у истоков, и для нашего английского слуха (притупленного десятилетиями лязга ткацких станков) его голос звучит загробно, как у Квентина Тарантино. Так что про «Джанго освобожденного» мы его спросить не рискнули. Зато он охотно поделился планами развития Django, рассказал о работе над функциями и о насущной задаче создания идеальной документации.

LXF: Похоже, за последние годы Django ничуть не сдал свои позиции. Как успехи?
ДжК-М: Прекрасно, прекрасно. Django растет взрывными темпами, и по-моему, темпы роста – одна из главных проблем сообщества. Столь быстрый рост действительно чреват проблемами. Это из серии ответов на тот злополучный вопрос «Какие у вас недостатки?» – нам бы ваши проблемы, да? Это интересно и увлекательно.

LXF: Помнится, вы пытались подсчитать число пользователей Django.
ДжК-М: Да, я занимался этим где-то раз в год.

О РАСШИРЕНИИ

«По-моему, темпы роста — одна из главных проблем нашего сообщества.»

LXF: И каким же методом?
ДжК-М: Да наугад [*смеется*]. Нет, на самом деле есть разные метрики. Google Trends стал кошмаром – имя Django теперь ассоциируется с фильмом Тарантино, так что поиск по нему не работает, да и музыкант такой есть. Но можно судить по другим вещам.

Число разработчиков, например, или создателей заплат, число электронных подписчиков. Есть примерное число скачиваний самого Django – по числу зарегистрированных IP, уровню трафика на сайте и тому подобное. Но есть, конечно, и доля анонимных пользователей, подмножество которых читает документацию, подмножество создает заплаты или участвует в развитии проекта... В любом случае, тенденция очевидна, и видя рост этих чисел, я думаю, можно уверенно утверждать, что растет и сообщество, даже не располагая точными данными.

LXF: И этот рост ускоряется?
ДжК-М: За конкретными цифрами я бы лучше зашел в свой блог, но как будто в промежутке от пре-релиза 1.0 и года два после выхода версии 1.0 наблюдался экспоненциальный рост. Сейчас темп, похоже, несколько сбавился, но рост все равно очень быстрый. Если не ошибаюсь, трафик сайта за прошлый год увеличился вдвое или около того. Полагаю, бурный рост по-прежнему налицо.

LXF: А сферы применения Django изменились со временем?
ДжК-М: Значительно. Сначала он довольно долго не покидал пределов новостных редакций, считаясь по определению «медийным» инструментом.

LXF: Мы читали об этом, но никогда не осознавали полностью, как это работает.
ДжК-М: Изначально он задумывался больше как издательская платформа. Некое решение для информационно-ориентированных сайтов – где обычно нет общения, комментариев,

опросов и прочего – только обычный издатель и множество пользователей.

LXF: Как в CMS?

ДжК-М: Верно, Django ведь и вышел из CMS. То есть это не CMS, но используется примерно так же. И хотя с самого первого дня это был набор для web-разработки, такое представление закрепилось, от него мы и стали отталкиваться.

За последние три-четыре года изменилось прежде всего то, что мы это представление преодолели, и Django больше не рассматривают как «медийный» проект, скорее как универсальную среду для web-разработки; и, как следствие, нет сейчас такой сферы, где бы Django не использовался. Есть, конечно, первая лига – те, кто делает Discus, Instagram и Pinterest, – создатели передовых соцсетей, но с другой стороны, я в этом году сотрудничаю с одной малоинтересной компанией, из Fortune 500, над внедрением Django в их средства разработки. Это как части одной большой палитры.

LXF: Почему же предприниматели обращаются к Django?

ДжК-М: Многие компании сейчас завершают использование первого поколения CMS или внутрисетевых инструментов, либо самим этим инструментам уже около 10 лет. Это те компании, что раньше всех стали предлагать сетевые приложения, те инструменты, которым сейчас 10, 12, 15 лет, и все это время они пользовались Java, .Net, PHP. Будущее Java и .Net сейчас весьма туманно. Сами компании плохо представляют, что будет с Java в контексте всей этой неразберихи с Oracle и что ждет Microsoft как бизнес, а Microsoft направляет своим разработчикам какие-то безумные путаные директивы, равно как и Oracle. Им нужен новый облик и новые средства разработки; они присматриваются к прочему инструментарию, и Python кажется уже не столь жуткой и странной технологией. Им пользуется Google, и ему уже 20 лет. Это уже проверенный старый ветеран. И вот теперь многие компании хотят сделать на него ставку, а если вы думаете работать с Python в Сети, Django будет лидером с большим отрывом. Так что я считаю его очевидным выбором.

О ПРОПРИЕТАРНОМ VS OPEN SOURCE

> «Получив равноценную альтернативу, люди все чаще выбирают open source.»

LXF: То есть причиной перемен стало в том числе падение интереса к Java/.Net?

ДжК-М: Я вижу бурный рост web-мира Python, а не только Django. То есть у Django все прекрасно, но есть еще Pyramid, который привлек массу новых пользователей, равно как и Flask. И речь вовсе не о том, что Pyramid, Flask и прочие отнимают у Django его долю на рынке – мы тоже растем. И это идет на благо всему миру Python.

LXF: Значит, все дело в Python?

ДжК-М: Мне кажется, Python превращается в гиганта. Вот все говорят об OpenStack – его считают прорывом. А что такое OpenStack? Это Python и Django. Все инструменты управления – это Python, весь пользовательский интерфейс – это Django. Так что любая компания, решившая развиваться в облаке, теперь может выбрать проприетарного производителя – пойти к Amazon, или пойти к RackSpace, а может выбрать open source – ведь он уже не такой жуткий и странный. А выбрав open source, они выбирают Python. Это уже не шаг в неизвестность, а очевидный выбор.

LXF: То есть Python теперь — признанный инструмент web-разработки?

ДжК-М: Мне кажется, дело еще и в том, что динамические языки в целом получили гораздо больше признания. Была же такая старая «утка», что на них невозможно написать надежную программу, потому что компилятор вас не спасет. Сегодня, имея достаточный опыт работы со статическими языками – с Java и .Net – люди уже удостоверились, что это не так. Их опыт работы с Python, с другой стороны, свидетельствует о том же.

Специфика своя, конечно, есть, метод работы иной. Но платформы и Java/.Net, и платформы Python уже настолько зрелые, что не являются неизвестными величинами. Они уже не новы. Известны все их подводные камни, и люди делают выбор осознанно. И получив явную и равноценную альтернативу, люди все чаще выбирают open source, и значит, Python.

LXF: А ваша роль изменилась?

ДжК-М: Три года назад я был новичком, а теперь – консультант по работе с Python и Django. Помогаю компаниям освоиться в этом сегменте. Типичные клиенты – те, кто только окунается в этот мир и нуждаются в советчиках.

LXF: Они приходят к вам, зная о Django лишь в общих чертах?

ДжК-М: Когда как. Некоторые клиенты обращаются к нам, уже успев поработать с Python и Django, и хотят, чтобы мы проверили, все ли они делают верно. Некоторые же только делают первые шаги и хотят получить совет, кого им нанимать и как обучать свой персонал, как быстро встать на ноги, помощь с построением первичной архитектуры и прочее.

LXF: Разве первичная архитектура все еще важна? Мне казалось, что Django в этом отношении не столь требователен.

ДжК-М: Думаю, одно из преимуществ Python как языка и Django как инструмента в том, что здесь шире почва для эксперимента. Более высокоуровневые языки лаконичнее, а значит, вы можете решать те же задачи меньшим количеством кода. А значит, и с рефакторингом хлопот будет тоже меньше. Так что да, пожалуй, первичные ошибки здесь не так фатальны.

LXF: А *MySQL* по-прежнему используется как внутренний сервер?

ДжК-М: Знаете, это довольно забавно – похоже, что после того, как Oracle купили *MySQL*, с ним случилось то же, что и с Java. Компании очень

боятся вот этой неуверенности в будущем. Я так вижу, что *PostgreSQL* тоже подстегнул Python.

Так уж вышло, что большинство разработчиков Django – поклонники *Postgres*, и пользуются в основном им; я и сам его предпочитаю, хотя это не важно. В свое время *Postgres* и *MySQL* отличались гораздо разительнее, чем сейчас, когда это просто две великолепные программы. Мне кажется, сложности *MySQL* носят характер управленческий, а не технический.

LXF: Как вам удалось создать такую хорошую документацию для Django?

ДжК-М: Отчасти благодаря тому, что Адриан (Головатый [Adrian Holovaty], со-основатель Django) по образованию журналист, а я – филолог, и мы оба всегда относились к писанине серьезно.

LXF: Но большинство разработчиков норовит только кодировать.

ДжК-М: Мы уже с самого начала решили не вводить ни одной функции без документации. Без нее и соответствующих тестов, заплатка не считается готовой. И это достаточно высокая планка.

Я считаю, что во всех сообществах тон задается сверху. И в Django была превосходная документация с самого первого релиза. Мы сделали это культурным эталоном. И привлекаем людей, которые с этим согласны.

LXF: Значит, и таких как мы — тех, кто сможет это прочесть и что-то для себя узнать.

ДжК-М: Об этом я думаю в первую очередь. Мне на самом деле кажется, что в последнее время мы увлеклись качеством, а надо бы сконцентрироваться на другом. Мы подошли к той точке, когда документации стало так много, что ее организация и поиск чего-либо стали проблемой – проблемой, которой прежде не было. Прекрасно, что задокументировано абсолютно все; но найти то, что нужно, теперь трудновато. К этой проблеме необходимо подойти очень серьезно и найти решение.

LXF: Как же создается документация?

ДжК-М: Есть такой инструмент, *Sphinx*, пришедший из документации Python. Изначально она писалась на Latex, а значит, внести свою лепту было довольно сложно – нужно было владеть Latex. Тогда-то они и создали *Sphinx*, и он великолепен. Это лучший инструмент для технической документации, мне известный.

ПРО ДОКУМЕНТАЦИЮ

«Мы уже с начала решили не вводить ни одной функции без документации.»

LXF: Разработчики сами создают документацию к своим функциям?

ДжК-М: Да, сами. Не обязательно тот же человек – иногда один создает код, другой пишет документацию, третий ее редактирует. Мы стараемся проявлять гуманность. Для многих членов сообщества английский – не родной язык, и мы успокаиваем их: «Не переживайте, запишите только самое основное». Я более чем уверен, что никто не опишет функцию лучше, чем ее создатель. Ну, вы понимаете. Когда кто-то другой пытается воссоздать ход вашей мысли, у него все равно получится хуже, но это вовсе не значит, что вам надо быть великим писателем. Достаточно зафиксировать главное, а мы продолжим.

LXF: Два года назад вы сказали, что одно из главных преимуществ Django — широкая аудитория разработчиков. Это все еще так? Не трудно ли теперь ею управлять?

ДжК-М: У нас были сложности с пополнением действующих разработчиков. Первоначально политика была такая: новые члены принимались с согласия существующих комитетов, но когда людей становится много, это уже не работает. Пять человек еще можно привести к консенсусу. Ну, 10, ну, 15. Так что мы это отменили и немного упростили правила приема, что оказалось правильным решением. Наша пирамида по-прежнему очень широка в основании, а наверху ощущается острая нехватка волонтерской силы, скажем так, работников последнего звена – тех, кто проверяет и регистрирует готовые патчи.

И с этим мы постоянно боремся – привлекая новых сотрудников, но, естественно, есть огромная масса людей, пишущих код, и много людей, его проверяющих.

LXF: Что определяет стратегию развития проекта?

ДжК-М: В сообществе Python бытует шуточка, что Адриан и я выступаем в роли бархатных диктаторов, так ведь? Диктовать мы можем, но стараемся никогда этого не делать.

LXF: Значит, у вас нет планов на новый релиз?

ДжК-М: Есть черновые идеи. Я знаю, чего бы мне хотелось, и имею представление о том, как это можно воплотить, но все зависит от самих исполнителей.

LXF: Что-нибудь из серии Python 3000?

ДжК-М: Думаю, Python 3 был более целенаправленным. Было четкое видение того, что это будет – причем не одного человека, не только Гвидо (Ван Россум [Guido Van Rossum] – бархатный диктатор Python), а сразу многих. Но было конкретное описание и направление, и идея в основе всего этого. Наш релиз менее жесткий. По принципу – все, что будет отлажено к оговоренному сроку, пойдет в релиз.

LXF: И вы не боитесь, что функции окажутся слишком разномастными?

ДжК-М: Мы очень следим за этим, особенно в последнее время. Мы теперь более скептически настроены к новым функциям. Как правило, мы задаем вопрос: «А почему нельзя сделать это сторонним внешним модулем?»

Мы хотим, чтобы люди доказали нам, что эта функция либо чрезвычайно важна абсолютно для всех и достойна войти в ядро, либо что ее оптимально сделать именно внутренней; если не вводить таких ограничений, наступит хаос. Приходится постоянно отсекать метастазы этих мелочных «нужных мне» функций. К сожалению, работа куратора продукта на 80 % состоит в том, чтобы говорить «нет». LXF

Планшетники: Последний рубеж

Джоно Бэкон, менеджер сообщества Ubuntu, демонстрирует, как запустить Ubuntu и рабочий стол Unity на планшетнике Nexus 7.

Большинство из вас, читающих эту статью, уже знакомы с Ubuntu, даже если вы им и не пользовались. Скорее всего, знакомство началось с настольного дистрибутива Ubuntu, который популярен с момента своего появления в 2004 году. Некоторые, возможно, знают Ubuntu благодаря серверам и облачным сервисам, и на сегодня Ubuntu является самой популярной операционной системой, используемой в «облаках».

Возможно, более широкий взгляд на Ubuntu не столь очевиден; это вездесущая операционная система, которая приносит конвергентность на многочисленные устройства и облачные сервисы.

Что же в данном контексте означает «конвергентность»? Цель Ubuntu – создать единую ОС, способную работать на любом из ваших устройств, поставляя на эти устройства устойчивый и простой пользовательский интерфейс.

> «При виде работы Unity на разных устройствах его смысл яснее.»

Этот устойчивый пользовательский интерфейс и был целью иногда спорного пользовательского интерфейса Unity, дебютировавшего на рабочем столе Ubuntu пару лет назад. При виде работы Unity на этих разных устройствах его смысл становится намного яснее. Ubuntu занялся оборудованием, когда представил Ubuntu для Android на CES в 2012. Это дало возможность использовать Android на телефоне, однако при присоединении к нему экрана, клавиатуры и мыши будет загружаться полноценный рабочий стол Ubuntu с интегрированными приложениями, контактами и сообщениями от Android внутри системы. За ним последовал Ubuntu для ТВ, который предоставлял полноценный интерфейс для управления ТВ и мультимедиа, разработанный специально для приставок и Smart TV. Свежее дополнение – это Ubuntu для телефонов, представленный в январе.

С настольного ПК – на устройство

Как установить Ubuntu на вашем Nexus 7.

При запуске Ubuntu на телефоне, планшетнике или ТВ используется одна и та же основа, но оборудование и требования к производительности для этих устройств совершенно разные. К примеру, при выпуске настольного Ubuntu мы производим образ общего назначения, пригодный для установки на целом ряде устройств. Если вам нужно запустить Ubuntu на телефоне, планшетнике или ТВ, то требования к оборудованию, как правило, более ограниченные и узкие. Здесь меньше целевых процессоров, и в основном это встроенные процессоры – чипы ARM. Есть также ограничения по памяти, часто между 512 МБ и 1 ГБ, и есть специализированные компоненты оборудования, такие, как сотовая радиотелефонная связь, акселерометры и другие, требующие поддержки.

С тех пор, как Ubuntu обратил внимание на эти устройства, предпринимались попытки оптимизировать платформу для эффективной работы. Одной из ключевых частей этой работы был выбор общеупотребительного целевого устройства и оптимизация системы внутри сообщества для этого устройства.

Unity отпугнул многих, но все это обретает смысл, когда вы видите его на устройстве с тачскрином.

Оптимизация для устройств

Планшетник Nexus 7 был выбран по нескольким причинам. Многие телефоны и планшетники являются закрытыми и не позволяют установить альтернативную операционную систему. В этих случаях любому желающему работать с альтернативной ОС требуется либо взлом устройства, либо заключение коммерческого соглашения с производителем по использованию устройства. Nexus 7 – устройство, легко поддающееся разблокировке, что делает его разумным выбором для нашего сообщества, которое сможет участвовать в подготовке Ubuntu к работе. Помимо этого, Nexus 7 также отличается достаточной мощностью, но не до такой степени, чтобы нам не пришлось потрудиться на ниве оптимизации Ubuntu. Кроме того, это устройство относительно недорогое и широко доступное.

Здесь важно подчеркнуть, что цель настройки Ubuntu для работы на Nexus 7 заключалась отнюдь не в создании планшетной версии Ubuntu. Целью было заставить настольный Ubuntu эффективно работать на этом устройстве, а надо помнить, что таковой разработан для клавиатуры, мыши и большого экрана, а не маленького тачскрина.

> «Те, что ожидают сексуальной планшетной версии, будут разочарованы.»

Команда Ubuntu прекрасно осознает, что пользовательский опыт работы с Ubuntu на Nexus 7 не является оптимальным; они собирались заставить самый низкий уровень, неграфические компоненты, такие, как ядро, сервер дисплея, сетевой стек и другие элементы, работать при небольшом объеме памяти и с низким энергопотреблением. Те, что ожидают сексуальной планшетной версии Ubuntu, будут разочарованы; данная статья – о том, как установить Ubuntu на устройстве совершенно иного типа, чтобы он *работал* – просто потому, что мы это можем.

Установка на Nexus 7

Чтобы установить Ubuntu на Nexus 7, вам понадобится (внимание, спойлер) Nexus 7. К счастью, вы можете использовать модели 8 ГБ, 16 ГБ или 32 ГБ. Важно отметить, что хотя вы всегда можете вернуться к системе Android, поставляемой с этим устройством, при переключении на Ubuntu или обратно на Android вы потеряете все свои данные и настройки. Поэтому сделайте резервные копии всех своих важных настроек.

Рекомендую просто поместить важные для вас данные в персональное облако Ubuntu One; существуют клиенты Ubuntu One для Ubuntu, Windows, Mac, iPhone и Android. Более подробную информацию об этом вы найдете в разделе «Используем Ubuntu One».

Используем Ubuntu One

Ubuntu One – это персональное облако, встроенное в Ubuntu, доступное также и для других систем. Оно предоставляет место для хранения ваших персональных файлов, музыки, видео, фотографий и настроек. Контент, размещенный вами в Ubuntu One, синхронизируется по разным вашим устройствам. Например, если вы сохранили в вашем персональном облаке Ubuntu One фотографию своего песика на скейтборде, она появится на всех ваших устройствах, соединенных с Ubuntu One. Ubuntu One предлагает 5 ГБ бесплатно, и вы можете прикупить дополнительное пространство. Это система для синхронизации ваших файлов и другого контента между вашими устройствами, работающими на Ubuntu.

Ubuntu One также включает такие функции, как публикация файлов, фотографий и другого контента онлайн, а также трансляцию потоковой музыки с вашей учетной записи Ubuntu One на ваши устройства. Например. если вы храните свою музыкальную подборку в Ubuntu One, вы можете передавать ее на свой телефон, не скачивая ее на телефон.

Ubuntu One изначально встроен в Ubuntu, но также доступен для Windows, Mac, Android и iPhone. Вы можете скачать программы для этих систем с http://one.ubuntu.com.

Работая в Ubuntu, вы можете зарегистрироваться в Ubuntu One из включенного приложения; просто поищите “ubuntu one” в Dash.

Nexus 7 (поставляемый с Android 4.1) запросто поддерживает несколько ОС.

Помимо Nexus 7, вам понадобится кабель MicroUSB. Его вы используете для соединения Nexus 7 со своим ноутбуком и установки программ оттуда на Nexus 7. Опционально вы можете использовать основной кабель для соединения мыши и клавиатуры с Nexus 7 через USB, но вероятнее всего, вы будете использовать тачскрин вместо мыши и экранную клавиатуру вместо физической. Для установки Ubuntu на Nexus 7 вам нужно, чтобы на вашем ноутбуке/настольном ПК была установлена Ubuntu 12.04 или более новая версия. Прежде всего вам надо будет установить PPA, содержащий программу установки Nexus 7. Для этого откройте терминал, щелкнув по кнопке Ubuntu, введя **terminal** в dash и нажав на значок.

Теперь введите следующие команды:

```
sudo add-apt-repository ppa:ubuntu-nexus7/
ubuntu-nexus7-installer
sudo apt-get update
sudo apt-get install ubuntu-nexus7-installer
```

Эти команды установят PPA, используемый разработчиками Nexus 7, обновят ваш список пакетов, а затем установят программу установки Nexus 7.

Теперь нужно разблокировать Nexus 7, чтобы мы могли сделать установку. К счастью, это простой процесс, и – помните: вы всегда можете вернуться к Android, если вы так решите, но потеряете свои данные, файлы и настройки. Так что позаботьтесь о резервных копиях. Подключите Nexus 7 к вашему ноутбуку через USB и перезагрузите Nexus 7, чтобы он показал программу загрузки Android. Для этого выключите устройство, нажав и удерживая кнопку питания, и выбрав выключение; устройство выключится. В этом выключенном состоянии нажмите и удерживайте кнопку уменьшения громкости и нажмите на кнопку питания. Продолжайте удерживать кнопки уменьшения громкости и питания, пока не увидите пользовательский интерфейс программы загрузки, который отобразит логотип Android со вскрытой крышкой и большую кнопку Start.

Затем запустите **fastboot** и убедитесь, что устройство есть в списке:

```
sudo fastboot devices
```

Вы должны увидеть следующее:

```
1234567891234567 fastboot
```

Теперь приступайте к разблокировке программы загрузки Nexus 7, запустив

```
sudo fastboot oem unlock
```

Появившийся экран Nexus 7 спросит вас, хотите ли вы разблокировать программу загрузки, и дополнит это неким текстовым предостережением. Нажмите на Yes для продолжения.

> «Кому одного пальца мало, Nexus 7 поддерживает мультитач.»

Завершите процесс, запустив

```
sudo fastboot reboot-bootloader
```

Для этого также можно нажать на кнопку питания. Когда Nexus 7 перезагрузится, вы должны увидеть надпись мелким красным шрифтом: LOCK STATE = UNLOCKED. Итак, вы сделали разблокировку и готовы установить Ubuntu!

Чтобы установить Ubuntu на Nexus 7, надо сначала обеспечить, чтобы никакие другие устройства не были подключены к вашему ноутбуку/настольному ПК через USB; подключен должен быть только Nexus 7. Теперь щелкните по кнопке Ubuntu, найдите “nexus” в Dash, и вы увидите в списке “Ubuntu Nexus7 Installer”. Щелкните по значку и следуйте инструкциям в появившемся диалоговом окне. Релиз Ubuntu для Nexus 7 будет скачан и установлен на устройстве. Теперь, когда Ubuntu установлен на вашем Nexus 7, можете перезапустить устройство, после чего загрузится рабочий стол.

Хотя вы, возможно, видите на своем плашетнике Ubuntu, помните, что основная цель проекта установки Ubuntu на Nexus 7 не в оптимизации рабочего стола для планшетника, а в том, чтобы заставить низкоуровневые компоненты операционной системы работать эффективно. Рабочий стол работает, но есть пара вещей, которые вам стоит знать при его использовании. Во-первых, сенсоры на планшетнике включены. Вы, например, можете использовать палец, чтобы перемещать Launcher в левой стороне экрана вверх и вниз, и стукнуть по экрану, чтобы запустить приложение.

В Nexus 7 поддерживается ряд жестов. Удар одним пальцем эквивалентен щелчку мышью, он используется для выделения и включения чего-то. Подобным же образом, пальцем можно перетащить объект, как если бы вы щелкнули по нему мышью и перетащили его.

В качестве более экзотического примера – двойной удар пальцем выделит слово или объект, а тройной – абзац или группу объектов.

Кому одного пальца мало, Nexus 7 поддерживает мультитач. Например, загрузите приложение так, чтобы на экране появилось окно. Теперь стукните по окну тремя пальцами. Вы увидите, что по краям окна и в середине появились значки. Можете использовать их для изменения размера окна, а значок в центре служит для поворота окна. Также можно ударить по экрану четырьмя пальцами – это сработает как быстрый запуск Dash.

Клавиатура

Когда на вашем Nexus 7 вместо специального пользовательского интерфейса для планшетника появится Ubuntu, у вас, вероятно, возникнет вопрос: а как набирать текст без клавиатуры?

К счастью, Ubuntu уже давно снабжен предустановленными функциями доступа, среди которых – экранная клавиатура. Если вы хотите видеть эти функции на обычном настольном Ubuntu, загрузите Ubuntu Settings и выберите раздел Universal Access. Здесь и включается клавиатура. На Nexus 7 она уже любезно включена по умолчанию, и должна появляться всякий раз, когда надо ввести текст. Если в силу каких-то причин

Команда Ubuntu (как мы воображаем) в поте лица трудится над официальной планшетной версией.

› Вы всегда при желании сможете вернуться к Android.

клавиатуры нет, стукните по значку на панели с четырьмя квадратами, выберите Show Onboard, и клавиатура появится.

Есть пара хитростей, которые могут пригодиться при использовании клавиатуры. Во-первых, значок X в углу позволяет убрать клавиатуру, когда она вам не нужна. Или можете использовать значок с четырьмя стрелками под кнопкой X для перемещения клавиатуры.

Клавиатура также предоставляет полезные функции взамен мыши. Например, трудно щелкнуть правой кнопкой мыши, если у вас нет ни мыши, ни правой кнопки. Чтобы имитировать щелчок правой кнопкой, сначала коснитесь кнопки со стрелкой в верхней правой части клавиатуры, и вы увидите дополнительные кнопки. Затем среди возникших кнопок выберите кнопку со стрелкой вправо (она должна появиться рядом с той кнопкой, которой вы только что коснулись). Теперь коснитесь области экрана, к которой вы хотели применить щелчок правой кнопкой мыши. Вы должны увидеть тот же эффект, как и от щелчка правой кнопкой (например, появление меню).

Если при работе с экранной клавиатурой произошли проблемы или ошибки, вы можете управлять Nexus 7 некоторыми другими способами. Первый и самый простой подход – использовать кабель MicroUSB Host (OTG). Подключите его и подсоедините клавиатуру.

Другой подход – применить *VNC* для управления Nexus 7. *VNC* – это сетевой протокол, используемый для экспорта экрана на другое устройство. Для начала запустите 'Vino' на Nexus 7 и в появившемся окне отметьте галочкой окошки 'Allow Other Users To View Your Desktop [Разрешить другим пользователям просматривать ваш рабочий стол]' и 'Require The User To Enter This Password [Требовать у пользователя ввести этот пароль]', а во втором из них введите пароль. Можете убрать отметки из других окошек, и внизу выберите Only When Someone Is Connected [Только когда кто-то подсоединен]. Закройте окно, и теперь у вас будет работать демон *VNC*.

Теперь вы можете использовать на своем компьютере программу просмотра удаленного рабочего стола для соединения с Nexus 7 и управления им. Возможно, в программе просмотра вам нужно будет отметить 'Use JPEG Compression [Использовать сжатие JPEG]'; это должно немного увеличить производительность.

Установка программ

Настроив свой Nexus 7, вы сможете использовать его, как обычный настольный Ubuntu. Единственная область, где надо быть поосторожнее, это установка программ. Вы должны использовать только программы в архиве и не включать архивы *-updates* и *-security*, если используете релиз более старый, чем 13.04. Это может нарушить обновления, созданные специально для Nexus 7.

Если вы на Nexus 7 используете 13.04, рекомендуем регулярно обновлять систему, скачивая отладки. Можете сделать это, загрузив терминал (касание четырьмя пальцами для доступа в Dash, введите **terminal** и щелкните по значку) и затем запустив

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

Когда вам предложат ввести пароль, по умолчанию это 'ubuntu'. Для установки программ можно также использовать Ubuntu Software Centre, который включен в систему,. Однако будьте осторожны при установке приложений – не от всего стоит ожидать ровной работы: что-то работать не будет, или будет, но не в соответствии с вашими ожиданиями. Например, на момент написания этой статьи некоторые игры (в частности, 3D) и приложения не отличаются хорошей работой на устройствах с ограниченными ресурсами вроде Nexus 7.

Возврат к Android

Если вы не хотите оставлять Ubuntu на Nexus 7 и хотите вернуться к Android, загрузите последнюю версию образа для вашего устройства с http://bit.ly/QDfbtq и сохраните ее на диск. Прежде чем заново установить Android, сделайте резервные копии всех настроек и файлов, которые были на Nexus 7. Теперь подключите свой планшет к ноутбуку/настольному ПК кабелем MicroUSB (убедитесь, что это единственное устройство, подключенное к ноутбуку/настольному ПК), перезапустите планшет и удерживайте кнопку уменьшения громкости, пока он перезагружается. Переведите устройство в режим fastboot:

```
sudo fastboot devices
```

Распакуйте скачанный tar-архив. Например: **tar zxvf nakasi-jro03d-factorye102ba72.tgz**. Потом запустите программу перепрошивки:

```
cd nakasi-jro03d/
sudo ./flash-all.sh
```

Вы, возможно, увидите предупреждения, которые можно игнорировать – восстановление будет продолжаться:

```
archive does not contain 'boot.sig'
archive does not contain 'recovery.sig'
archive does not contain 'system.sig'
```

Затем можете снова заблокировать устройство после еще одной загрузки в режиме fastboot:

```
sudo fastboot oem lock
```

Теперь Android будет установлен и готов к работе!

Если слой рабочего стола Ubuntu не работает так, как вы ожидали, не удивляйтесь, что разработчики не очень-то этим интересуются; они сконцентрировались на ядре, X-сервере и прочих низкоуровневых компонентах.

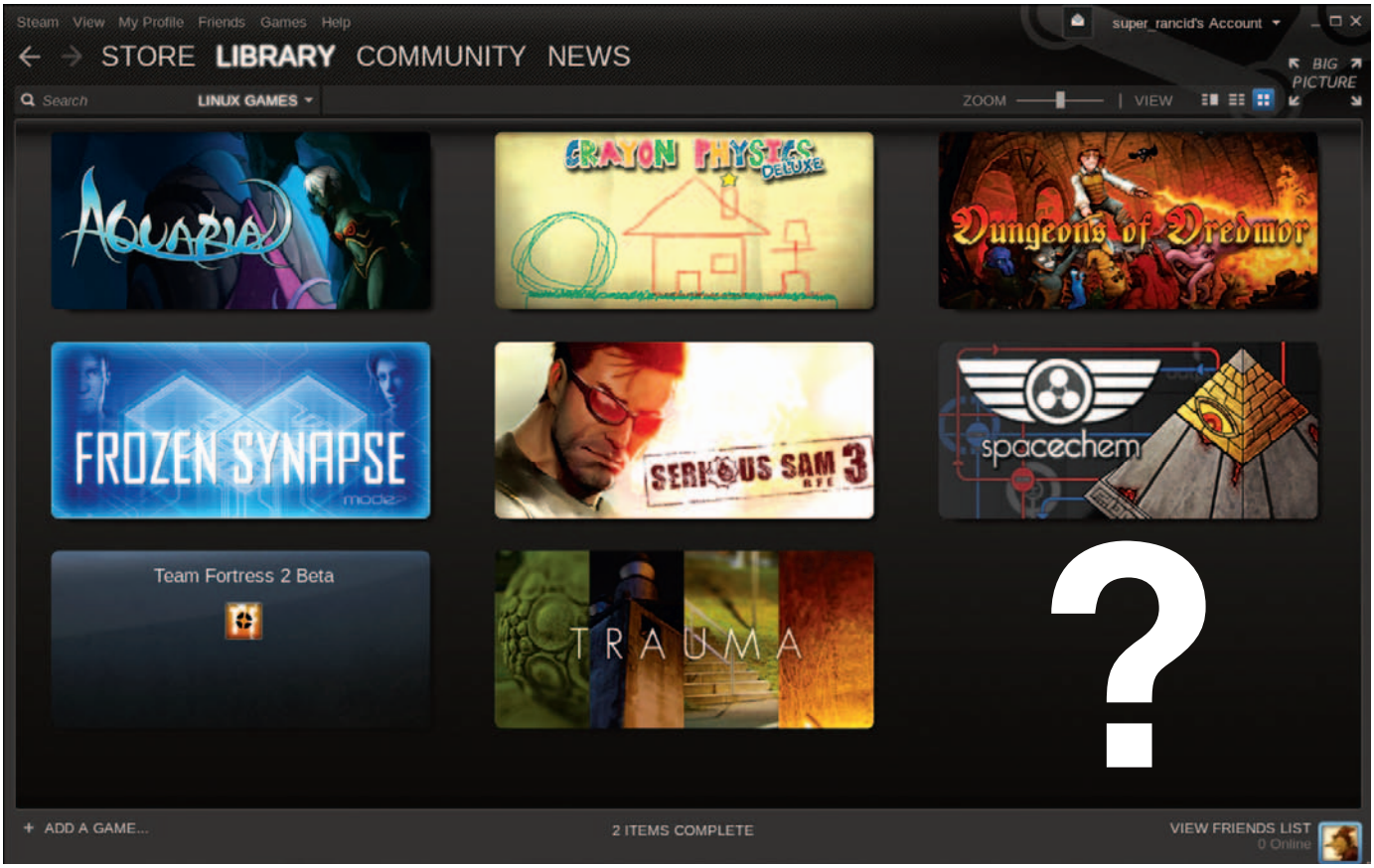
Тем не менее, наслаждайтесь Ubuntu на Nexus 7 и обязательно сообщайте об ошибках, если обнаружите их на низком уровне – для этого запустите команду **ubuntu-bug** в терминале. Удачи! LXF

Одна ОС, чтобы всех их единою властью связать

За последние годы Ubuntu захватила весьма солидное количество устройств. На всех устройствах Ubuntu работает одинаково, просто делая поправку на форм-фактор.

› Ubuntu для ТВ предоставляет плейер мультимедиа и интерфейс пользователя Smart TV для просмотра и записи ТВ и других видов мультимедиа.

› В супермощных телефонах Ubuntu для Android позволяет работать с полноценным настольным Ubuntu прямо с телефона.



Игры в Linux: На всех парах!

Маянк Шарма празднует взятие настольным Linux последнего бастиона. Присоединяйтесь!

Отсутствие в Linux качественных игр, достойных массовой популярности — любимый аргумент критиканов Linux как серьезной настольной ОС. Рады сообщить, что разработки последних месяцев камня на камне не оставили от этих отговорок.

Если у Linux до сих пор и была ахиллесова пята, то это – игры. Конечно, качественные игры с открытым кодом есть, но их гораздо меньше, чем в Windows. Однако ситуация должна измениться в связи с намерением Valve Corporation, разработчика ряда популярнейших игр, обратиться к Linux. Гейб Ньюэлл [Gabe Newell], главный босс Valve, открыто выразил свою поддержку Linux, и наконец-то компания вложила деньги в свой интерес и выпустила рабочую бета-версию своего клиента Steam [*англ.* паровик] для Linux, вместе со множеством игр.

Пользователи Linux уже давно играют в игры на Steam благодаря приложениям, обеспечивающим совместимость, типа *Wine* или *PlayOnLinux*. Однако нам очень не хватало родного клиента Linux и официальной поддержки нашей платформы от Valve. По слухам, бродящим в Интернет, в Valve уже приступали к работе над клиентом Linux. Однако ускорила события вовсе не растущая популярность нашей ОС у пользователей рабочего стола. Переломом стал последний релиз от Microsoft – Windows 8, который Ньюэлл считает «катастрофой для всех в сфере ПК».

> «Переломом для Valve стал последний релиз от Microsoft — Windows 8.»

Выступая на игровой конференции Casual Connect в июле 2012 года, Ньюэлл – бывший сотрудник Microsoft – поделился своими опасениями по поводу интеграции *Windows 8 Store* в последний релиз Microsoft и ее последствиями для дохода Valve. Он полагает, что после запуска Windows 8 потребители и разработчики соблазнятся официальной платформой дистрибутива Windows, особенно с учетом таких функций, как интеграция Xbox LIVE, и отойдут от Valve Steam.

Спасая свои доходы, Ньюэлл выбрал Linux как стратегию хеджирования. Вскоре после конференции Valve

Steam в Linux: Факты

Steam – это система доставки контента, разработанная Valve Software. Две известнейших игры от этой компании – *Half-Life 2* и *Counter-Strike*, и они основаны на ее игровом движке Source. Эти игры, как и игры от других студий и разработчиков инди-игр, предлагаются через клиент Steam. В прошлом пользователи Linux играли через клиент Steam для Windows, используя совместимость через такие приложения, как *Wine*. В ноябре Valve объявила о бета-версии родного клиента Steam для Linux. Этот клиент поддерживает бесплатную игру *Team Fortress 2*, и к ней постоянно добавляются новые игры. Бета-клиент включает Big Picture, новый режим для использования с ТВ.

› На данный момент, Ubuntu — единственная платформа, поддерживаемая Linux Steam beta.

объявила о создании команды разработчиков для переноса на Linux клиента Steam и таких популярных игр, как *Left 4 Dead 2*.

Через несколько месяцев лихорадочной работы, в ноябре 2012, компания объявила о выпуске Linux-клиента Steam в бета-версии с ограниченным доступом. Клиент поддерживал множество популярных игр, таких, как бесплатная *Team Fortress 2*, и предлагал еще пару дюжин на продажу. Потом, как раз к Рождеству, Valve открыла бета-версию Steam для всех, с расширенной библиотекой из более чем 40 игр. На данный момент Valve официально поддерживает клиент Steam только на Ubuntu, чтобы обеспечить клиенту прочное основание. Однако впоследствии она распространит свои попытки на другие дистрибутивы.

> «Спасая свои доходы, Ньюэлл выбрал Linux для хеджирования.»

Клапаны открыть!

Valve [*англ.* клапан] – не единственная игровая компания, обратившая внимание на Linux. Подобный же шаг предпринимался Blizzard, разработчиками невероятно популярной игры *World of Warcraft*. Хотя в прошлом компания не особенно рвалась портировать свое добро в Linux, Майкл Ларабел [Michael Larabel] из Phoronix сообщает о свежих разработках Blizzard, нацеленных на Linux. Как сообщают его источники в компании, Blizzard этим летом официально сообщит о портировании как минимум одной из своих игр в Linux.

Еще одна крупная студия, которая может объявить о выходе игр под Linux – THQ. Недавно компания выпустила ряд своих игр в Humble Bundle, но в отличие от других, они не кросс-платформенные и доступны только через Steam. Кроме $5 000 000, этот пакет игр чисто для Windows принес и мощную обратную связь от пользователей Linux. По сути, вполне достаточную, чтобы заставить THQ пересмотреть свою стратегию по Linux.

Президент компании, Джейсон Рубин [Jason Rubin], опубликовал в Твиттере: «Получили внятное послание от Linux через обратную связь #HumbleBundle. По ходу, оцениваем соотношение расходов и прибыли». В последующем интервью для Polygon.com он разъяснил, что «информация, которую я извлек из кучи твитов и комментариев по поводу продаж THQ на Humble Bundle, такова, что налицо энергичное сообщество геймеров, которые применяют другие ОС помимо главенствующих, и такая компания, как THQ, не должна их игнорировать».

Valve (а скоро и Blizzard, и, будем надеяться, THQ) присоединяется к Humble Indie Bundle, которая возглавила дело портирования игр в Linux, как мы рассказывали в нашем материале в **LXF167**.

Но портирование игровых движков важнее, чем портирование отдельных игр, и 2012 год стал свидетелем боевых действий и на этом фронте тоже.

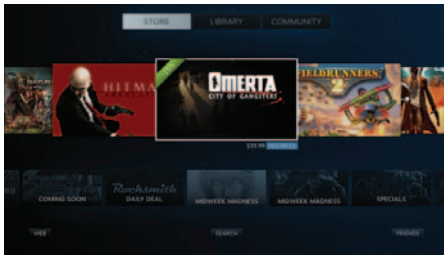
Игровой движок – это центральная библиотека, которую разработчики используют для определения аспектов игры, таких, как графика, звук, методы ввода, сетевые ресурсы и другие системы. Поскольку в игровом движке нет специфического для определенной игры кода, один и тот же движок можно использовать для разработки разных игр. Есть множество игровых движков с открытым кодом. Среди самых популярных – Panda3D, применяемый различными разработчиками, включая Диснея, для таких игр, как *Pirates of the Caribbean Online* и *Ghost Pirates of Vooju Island*. Еще есть ioQuake3, являющийся продвинутой версией движка Quake 3, и на нем работают такие игры, как *Turtle Arena*. Еще один хорошо известный движок – Cube, который используется в *Cube 2: Sauerbraten* и *Red Eclipse*.

› Некоторые игры Humble Indie Bundle были настолько хороши, что мы рассказывали о них в своих обзорах.

Коммерческие возможности

Помимо движков с открытым кодом, есть множество весьма достойных коммерческих. Valve вместе со своим клиентом Steam портирует в Linux также и свой движок Source. Движок Steam используется не только в играх Valve – его применяют и разработчики инди-игр, такие, как компания The Chinese Room из Брайтона, которая употребила его в своей игре *Dear Esther* и добилась наград. Разработчики *Dear Esther* ищут пользователей Ubuntu для тестирования своей игры на этом дистрибутиве.

Раз уж движок Source портируется в Linux, мы можем твердо надеяться, что за ним последуют новые разработчики, принося свои игры в Linux. Свои движки в Linux портируют и другие. Разработчики игрового движка Unity также горят энтузиазмом насчет Linux. Unity 4 ввела поддержку Linux, предъявив играбельные демо-версии двух игр, *AngryBots* и *Unitroids*. А еще есть *Rochard*, одна из первых игр, созданных на движке Unity 3D, который работает на Linux. Он вышел в *Humble Bundle 6*. Затем вышел *Humble Bundle 7*, »



› Вы увидите еще большие чудеса от режима Steam Big Picture по мере его появления на консолях Valve, работающих на Linux.

Феномен Humble Bundle

В декабре 2008 Джефф Розен [Jeff Rosen] из Wolfire Games заявил, что имеет смысл поддерживать операционные системы, до тех пор игнорируемые игровой индустрией. Полтора года спустя он и четверо других разработчиков скоординировали первую в своем роде акцию в мире игр. Помимо использования схемы «плати сколько не жалко», главным коммерческим аргументом было то, что все игры работали на нескольких ОС – Windows, Mac и Linux. Humble Indie Bundle был хорошо принят пользователями, и Розен запустил компанию Humble Bundle Inc, которая фокусировалась на предложении кросс-платформенных игровых пакетов.

Компания опубликовала более дюжины пакетов, соержащих более ста игр.

Desura

Если, по-вашему, Steam – единственный сервис по распространению игр для Linux, подумайте еще раз. Платформа Desura очень похожа на Steam, и предлагает несколько игр для Linux, в том числе коммерческие и игры с открытым кодом. Фактически, недавно сам мультиплатформенный клиент Desura вышел под GPL.

Большинство игр на Desura разработаны создателями инди-игр. Помимо игр, платформа также предоставляет патчи для игр, игровые режимы и дополнения. Сервис поддерживает социальные функции, такие как чат и блоги, и предлагает широкий выбор инструментов для разработчиков.

Чтобы установить Desura на свой дистрибутив, скачайте клиент (www.desura.com/desura-i686.tar.gz), распакуйте архив и дважды щелкните, чтобы запустить исполняемый скрипт. Он найдет другие нужные файлы в Интернете и установит клиент на ваш дистрибутив.

› **Вы найдете *PlayOnLinux* в репозиториях большинства дистрибутивов рабочего стола.**

› **Помимо ряда недорогих игр, Desura предлагает также и бесплатные.**

включивший первую родную игру Linux, созданную с использованием Unreal Engine 3 – *Dungeon Defenders*. На Unreal Engine от Epic Games работают такие популярные игры, как *BioShock*, *Batman: Arkham City* и серия *Mass Effect*. Есть также сообщения о том, что Overhaul Games работает над переводом в Linux *Baldur's Gate Enhanced Edition*. Игра работает на движке Infinity Enhanced Engine.

У некоторых разработчиков более амбициозные планы. Garage Games открыли исходный код своего движка Torque 3D в сентябре 2012. С тех пор появилось более 300 ответвлений кодовой базы, но все – для платформы Windows. Поэтому разработчики запустили кампанию на Indiegogo, собирая $30 000 для финансирования порта. После портирования разработчики смогут писать игры на Torque 3D полностью для Linux. А намерения Valve насчет портирования в Linux клиента Steam и того масштабнее: они объявили о своих планах крупного вброса в игровой сегмент через Linux.

> «Режим Big Picture предназначен для крупных экранов, типа ТВ.»

Крупным планом

Одна из интересных функций бета-клиента Steam – режим Big Picture [Большой формат]. Он предназначен для больших экранов, типа телевизионных, и является частью стратегии Valve по применению клиента Steam в консольном ПК, пригодном для гостиной. Идея заключается в том, что вы переключаете режим в своем клиенте Steam, затем подключаете контроллер USB к ПК, а ПК – к своему телевизору с высоким разрешением через кабели HDMI. Но прежде чем Valve затеет игры с оборудованием, ей нужна платформа, которую она сможет настроить под свои цели.

И нам всем известно, что только одна платформа допускает такую настройку. В интервью для *The Verge* Ньюэлл подтвердил, что их пакет для ПК, который часто называют Steambox, будет работать на Linux. За кулисами компания также работает с Nvidia, AMD и Intel над созданием лучших драйверов для Linux, и, похоже, это окупается. По словам компании, собственные игры Valve в Linux работают быстрее, чем в Windows.

Кстати об игровых консолях: теперь можно оформить предварительный заказ на OUYA, консоль на Android. Консоль стоимостью $99 получала финансирование исключительно через Kickstarter, и сорвала более миллиона долларов в первый же день кампании. Она будет работать на четырехъядерном процессоре Tegra 3 с 1 ГБ ОЗУ и 8 ГБ внутренней флэш-памяти. Разработчики OUYA наладили связи с OnLive, компанией поставки игр, и объявили, что в консоли будет встроенная игра в дополнение к *Final Fantasy III*. Ждите еще больше игр, когда начнутся поставки консоли, поскольку она еще может использоваться как инструмент разработки новых игр.

Не будучи особо разборчивыми насчет ПО, игры все же исторически способствовали популярности платформы, а также тому, чтобы производители оборудования улучшали поддержку и драйверы, что в свою очередь вело и в другие важные области компьютерной техники. Valve работает с Nvidia над написанием лучших драйверов для их видеокарт под Linux, а другие разработчики совершенствуют прочие аспекты рабочего стола Linux, чтобы сделать его лучшей игровой платформой и в конечном итоге повысить производительность для всех пользователей.

Помимо Linux-игр

И не забывайте, что Steam – уже не только игры. В октябре 2012 Valve начала публиковать в Steam неигровые программы. Она предлагает ограниченный выбор программ, таких как *ArtRage Studio Pro*, *CameraBag 2*, *GameMaker: Studio*, *3D-Coat* и *3DMark 11*, и они доступны только для Windows. Но если игровые студии видят в Linux своих пользователей, и, что еще важнее, добавочный поток доходов, разработчики приложений, без сомнения, тоже сумеют это разглядеть.

Эти разработки, вместе с отчетами корпоративных пользователей, обдумывающих возможности отказаться от Windows 8, дали всей экосистеме с открытым кодом еще один толчок в сторону рабочего стола Linux.

› ***Cube 2: Sauerbraten* — одна из лучших стрелялок с открытым кодом в ваших репозиториях. Попробуйте!**

Не хотите ли PlayOnLinux?

Приложение совместимости *Wine* – это ваш шанс, если вам нужны программы и игры Windows в Linux. Но чтобы *Wine* работало как надо, его следует настроить. Если вам не хватит умения (или мотивации), пусть за вас постарается PlayOnLinux.

PlayOnLinux можно рассматривать как графический интерфейс для управления Wine. Это приложение имеет солидный список приложений и игр Windows, которые можно установить одним щелчком, если у вас есть их исполняемые файлы или файлы setup. *PlayOnLinux* справляется со всеми сложностями управления *Wine*, и настраивает его так, чтобы оно соответствовало приложению, которое вы хотите запустить, обеспечивая нормальную работу с приложениями Windows.

В *PlayOnLinux* имеется множество игр и патчей, таких, как *Age of Empires*, *Counter-Strike*, *Portal*, *Mass Effect* и прочие, плюс другие популярные приложения Windows, такие, как *iTunes* и *Microsoft Office*.

PlayOnLinux уже вошел в репозитории большинства популярных дистрибутивов рабочго стола. Вы также можете найти пакеты последних версий для популярных дистрибутивов, таких, как Ubuntu, Fedora, Debian, Arch и т.д., на его сайте.

Установите Steam на Ubuntu

Следуя нашим инструкциям, установите бета-версию клиента Steam.

Теперь, когда бета-версия клиента Linux Steam открыта для общественности, вы тоже можете помочь Valve довести до блеска финальный продукт. Вам нужен процессор, более новый, чем Pentium 4 1 ГГц, или AMD Opteron с более 512 МБ ОЗУ, и свободные 5 ГБ на жестком диске.

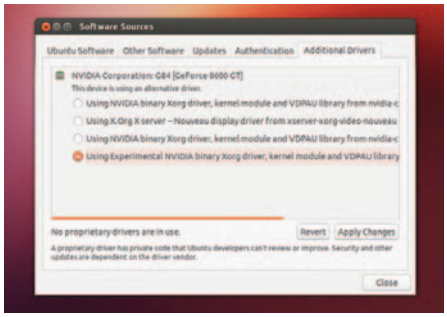
Чтобы насладиться современными играми, у вас должна быть достойная видеокарта. Касательно Valve, если вы – пользователь Nvidia, у вас должна быть видеокарта Series 6 или новее. Пользователям AMD рекомендуется Series 5 или выше, хотя более старые карты – такие, как HD 2400 Pro – тоже должны работать.

Если у вас нет ничего из вышеперечисленного, вам нужна, по крайней мере, мощная встроенная графика, например, Intel HD 3000 или новее.

Но что вам действительно понадобится – это хорошее интернет-соединение. Большинство игр, доступных для скачивания на Steam для Linux, имеют размер в несколько гигабайт, и большинство из них – многопользовательские, что требует наличия активного интернет-соединения: телефонный модем просто их угробит.

Также официально бета-версия клиента поддерживает только Ubuntu 12.04 и более новые дистрибутивы – хотя многие дистрибутивы, такие как Fedora, Fuduntu, Arch и другие, настроили свои неофициальные репозитории с бета-версией клиента Steam.

Шаг за шагом: Начните работу в Steam



1 Обновите драйверы

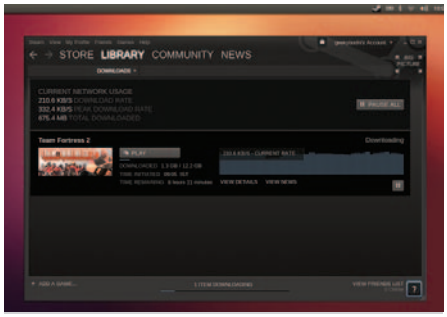
Убедитесь, что у вас есть самые свежие драйверы для вашего графического оборудования. Чтобы добиться наилучшей производительности от более новых карт, вам придется установить их проприетарное ПО. В Ubuntu 12.10 запустите Software Sources, щелкните по вкладке Additional Drivers и установите самые свежие экспериментальные драйверы. Вы можете пропустить этот шаг, если используете графику Intel.

2 Установите клиент

Перейдите на сайт Steam (http://store.steampowered.com) и скачайте Deb-пакет объемом 1,5 МБ. Можете дважды щелкнуть по файлу и установить его с помощью Ubuntu Software Centre, или установить его из терминала с помощью **sudo dpkg -i steam.deb**. Запустится графическая программа установки, которая затем скачает клиент.

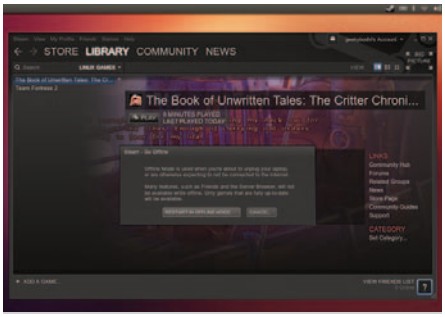
3 Создайте учетную запись

После установки клиент Steam попадет в Launcher. Когда вы запустите клиент, он даст вам возможность создать учетную запись. Если таковая у вас уже есть, введите свой логин. В этом случае, поскольку вы будете входить с другого компьютера, Steam отправит вам сообщение по электронной почте с подтверждением кода для авторизации нового клиента.



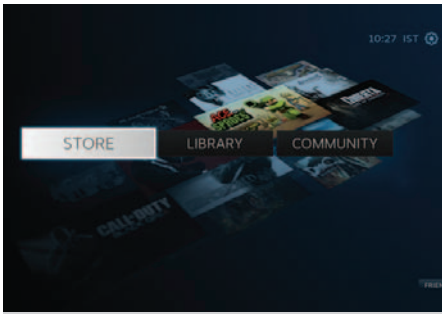
4 Скачивайте игры

Вот оно! Вы вошли в клиент Steam. Клиент Linux ничуть не отличается от клиента Windows. Вкладка Linux в разделе Store перечисляет все игры, доступные на настоящий момент для Linux. Вы можете для начала скачать *Team Fortress 2* – она доступна бесплатно, а затем приобрести дополнительные игры. Для некоторых игр предлагаются бесплатные демо-версии.



5 Режим оффлайн

Помимо многопользовательских игр, Steam также предлагает однопользовательские, в которые можно играть оффлайн. Steam запустится оффлайн при отсутствии интернет-соединения, а можно переключиться в него вручную, перейдя в Steam > Go Offline из самого клиента. Прежде чем играть оффлайн, надо обеспечить активацию игры, перейдя в режим онлайн хотя бы один раз после установки.



6 Соединитесь с ТВ

Valve также проводит бета-тестирование нового режима – использование клиента Steam на HDTV. Чтобы переключиться на этот режим, соедините свой компьютер с HDTV, запустите клиент Steam и щелкните по кнопке Big Picture. Помимо ссылок на магазин Steam, библиотеку и сообщество, в этом режиме имеется также web-браузер. Он разработан так, что навигация в нем осуществляется через игровой контроллер, но ладит и с мышью. LXF

Репозитории: Храним пакеты

Андрей Пономаренко и **Владимир Рубанов** исследуют тонкости поддержки пакетных репозиториев в дистрибутивах Linux.

Ключевым направлением развития любого современного дистрибутива Linux является предоставление пользователям наибольшего разнообразия программ на любой вкус. Для быстрой и удобной установки программ в дистрибутивах используются RPM- или Deb-пакеты, которые хранятся в специальных репозиториях на удаленных серверах в интернете или на установочном диске. Каждый пакет представляет собой заранее собранную из исходных кодов программу, готовую к использованию. Количество пакетов в репозитории может быть более 10000! Сборкой же пакетов занимаются так называемые мейнтейнеры дистрибутива. Это могут быть как энтузиасты из мирового сообщества, так и штатные специалисты компании – поставщика дистрибутива. И нелегок их труд.

Виды репозиториев

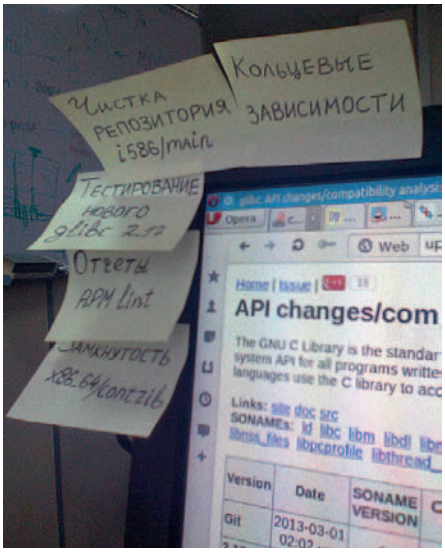
Общее количество программ в типичном современном дистрибутиве Linux – порядка десяти тысяч. Большинство из них хранятся в репозиториях, и только часть наиболее востребованных программ идет на установочных дисках. Для создания, сборки и дальнейшего обновления такого большого количества пакетов отводятся существенные человеческие ресурсы. Тем не менее, на поддержку всех пакетов в репозиториях, как правило, ресурсов все равно не хватает. По этой причине репозитории разделяют на две фундаментальные части: главный (main) и второстепенный (contrib). За каждым пакетом в главном репозитории должен быть закреплен определенный мейнтейнер, который будет следить за его состоянием. Пакеты же из второстепенного репозитория поддерживаются не постоянно – любыми членами сообщества дистрибутива, у которых появляется на это свободное время.

Выделяют также еще два вида специализированных репозиториев: для программ с закрытым кодом (non-free) и нарушающих патенты некоторых стран (restricted). Пакеты в этих репозиториях, как и в главном репозитории, должны поддерживаться явно назначенными мейнтейнерами, чтобы избежать возможных проблем по распространению дистрибутива.

Стоит заметить, что конкретные названия репозиториев могут различаться в разных дистрибутивах. Например, main-репозиторий в Fedora называют core-репозиторием, а restricted-репозиторий в дистрибутиве Mageia называют tainted-репозиторием.

В каждом из перечисленных репозиториев есть специально выделенный подрепозиторий для обновлений пакетов. Если пользователь хочет получать обновления системы, то он должен подключить эти репозитории в пакетном менеджере.

Чтобы можно было быстро считывать основную информацию о доступных пакетах, в репозиториях хранятся так называемые метафайлы. В них содержится информация о названиях, версиях, зависимостях и содержании пакетов. Исторически формат таких файлов различается в разных дистрибутивах. Например, в дистрибутиве ROSA это hdlist-файлы, а в Fedora – repodata-файлы.



Рабочий компьютер мейнтейнера буквально щетинится неотложными задачами.

Правила пакетирования

Чтобы поддерживать структуру дистрибутива и упростить процесс разработки, вводятся специальные правила (policy) для создания и обновления пакетов. Например, они описывают пути установки разных типов файлов программы в системе, правильный выбор имен пакетов в зависимости от их содержимого, и т. д. Все это в дальнейшем позволяет легче ориентироваться в системе и работать продуктивнее.

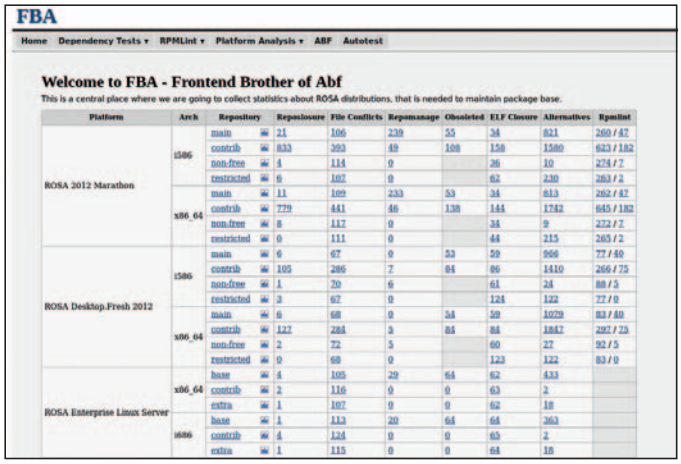
Automated Build Farm (ABF) — визуальная система сборки пакетов в дистрибутиве POCA.

Тестирование пакетов

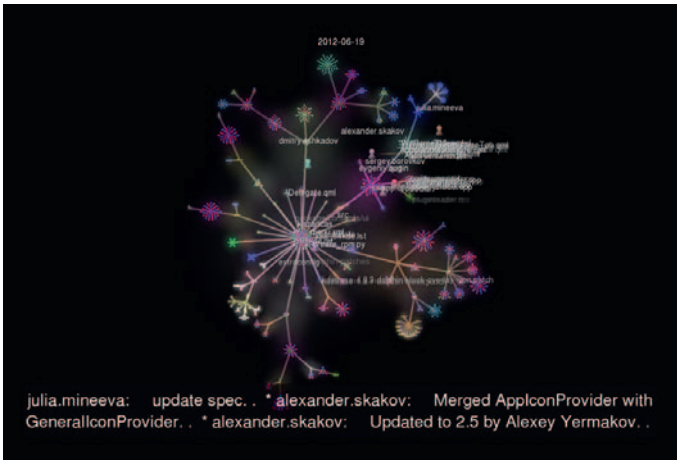
К сожалению, не все процессы по созданию пакетов достаточно автоматизированы, чтобы не возникало ошибок в пакетах. Для их поиска используются автоматические тесты, проверяющие соответствие содержимого пакетов установленным политикам. Одним из наиболее известных и универсальных инструментов для тестирования RPM-пакетов является *Rpmlint* [1]. Этот инструмент проверяет содержимое и атрибуты пакетов на соответствие нескольким сотням правил. Правила могут быть отключены, дополнены или изменены, чтобы соответствать политике конкретного дистрибутива. В зависимости от количества нарушенных правил пакету присваивается определенный уровень негодности (badness). Если негодность пакета больше некоторого заранее определенного уровня, то пакет необходимо исправлять. Допустимый же уровень негодности можно варьировать в зависимости от доступного количества мейнтейнеров, сроков выпуска дистрибутива и количества поддерживаемых пакетов. Аналогом данного инструмента для дистрибутивов, использующих Deb-пакеты, является *Deblint* [2]. Для репозитория главное, чтобы все пакеты в репозитории собирались, устанавливались и работали.

Замкнутость репозитория

Важнейшимм требованием к пакетным репозиторям является возможность безошибочной установки всех пакетов на компьютер пользователя.



› FBA — система контроля качества пакетов, применяемая в репозиториях дистрибутива ROCA.



› Визуализация процесса разработки репозитория с помощью CodeSwarm.

Проблемы с установкой пакетов могут возникать по разным причинам – например, если один пакет требует для работы несколько других, при этом одного из них по какой-то причине нет в репозитории. Такое явление называют неразрешенной зависимостью. Причиной их возникновения может служить как ошибка мейнтейнера, так и сбой в применяемых инструментах. Для контроля таких ошибок используются статические анализаторы замкнутости репозитория, т. е. наличия всех пакетов, требуемых другими пакетами. К сожалению, в силу существенных различий форматов репозиториев у разных дистрибутивов пока еще нет универсального инструмента для проверки замкнутости. Например, в таких дистрибутивах, как Mandriva или ROSA, используется инструмент *urpm-repoclosure* [3], а в Debian и Ubuntu используется инструмент под названием *debcheck* [4].

Конфликты по файлам

Еще одной причиной, препятствующей установке пакетов, может быть конфликт по файлам с другими установленными пакетами, т. е. когда один пакет пытается установить файл с тем же именем и по тому же пути, что и другой пакет. Иногда такая ситуация бывает допустимой – например, когда два пакета предоставляют две разные реализации виртуальной Java-машины или две разные реализации одной из стандартных библиотек. В таких случаях наличие конфликта должно быть отражено в свойствах пакета. Иначе конфликтующие файлы следует переименовать, обусловив корректную одновременную установку пакетов.

Кольцевые зависимости

Немаловажной проблемой пакетных репозиториев также являются кольцевые зависимости, т. е. зависимости вида A → B → C → A. Опять же, такие зависимости могут быть допустимы, когда одному пакету действительно требуется для работы другой, а тому нужен первый, но их нельзя поместить в единый пакет из-за политики именования и разделения программ на пакеты. В других случаях это может быть индикатором неправильно описанных зависимостей между пакетами. Причиной тут может быть не только человеческий фактор, но и некорректная работа автоматического генератора зависимостей у используемого пакетного менеджера. Это приводит к большим дополнительным временны́м затратам установщика пакетов и, возможно, к неправильному порядку установки пакетов из кольца.

Кольцевые зависимости можно находить с помощью специальных инструментов анализа. Например, в дистрибутиве ROSA для этого используется инструмент *urpm-repograph* [5].

Чистка репозитория

Ресурсы на поддержку пакетов ограничены, поэтому надо уметь определять наиболее важные. Со временем пакеты в репозиториях начинают устаревать, и от них надо постепенно избавляться. У некоторых пакетов появляются новые версии, и старые надо удалять. Другие же могут и вовсе стать ненужными в репозитории. К таким относятся, например, вспомогательные пакеты для пользовательских приложений. Процесс избавления от них происходит постепенно. Если какому-то пакету становится не нужен данный вспомогательный пакет, то он отмечается как устаревший (obsolete) в его свойствах. Затем такие же изменения производятся для остальных пакетов. Если же данный пакет будет помечен как устаревший во всех зависящих от него пакетах, то можно приступать к его удалению.

Популярность пакетов

Из-за ограниченности человеческих ресурсов, выделяемых на поддержку пакетных репозиториев, перед мейнтейнерами всегда стоит задача выбора набора пакетов, которым надо уделять внимание в первую очередь. Наиболее важные пакеты определяются либо по количеству других пакетов, зависящих от данного (обратных зависимостей), либо по количеству пользователей, использующих данный пакет (популярности пакета). Количество обратных зависимостей всегда можно подсчитать на основе анализа только репозитория. Для оценки же популярности пользователям предлагают установить дополнительный пакет, который будет сообщать разработчикам об используемых ими программах. Например, в системе Debian этот пакет называется *popcon* [6].

Заключение

Понимая перечисленные в данной статье возможные проблемы пакетных репозиториев, можно существенно сократить количество неприятностей с разработкой и использованием различных пакетов.

Именно поэтому для новых членов сообщества и мейнтейнеров разработчики дистрибутивов часто создают вводные Wiki-страницы (см. например [7] в ROSA или [8] в Debian), на которых собрана информация о правилах работы с пакетами в данном дистрибутиве, использовании инструментов, автоматических тестах, возможных ошибках и способах их предотвращения и исправления. Такие знания наряду с собственно средствами автоматизации процессов разработки и тестирования помогают существенно улучшить качество дистрибутива. LXF

Полезные ссылки

1. http://rpmlint.zarb.org/
2. http://www.debian.org/doc/manuals/programmer/ch-tools.html
3. http://wiki.rosalab.ru/en/index.php/Urpm-tools#urpm-repoclosure
4. http://qa.debian.org/debcheck.php
5. http://wiki.rosalab.ru/en/index.php/Urpm-tools#urpm-repograph
6. http://popcon.debian.org/
7. http://wiki.rosalab.ru/en/index.php/ROSA_Developer_QuickStart
8. http://www.debian.org/doc/manuals/developers-reference/

От XFree86 к X.org

Алексей Федорчук продолжает историю свободных воплощений оконной системы X.

Хронологически привязанную часть прошлой статьи (**LXF168**) я закончил на появлении в августе 1994 года XFree86 версии 3.0 – той самой, которой суждено было со временем лечь в основу десктопного Linux'а. В нынешней же статье предлагаю посмотреть, что было со свободной реализацией Иксов дальше.

Между «тройкой» и «четверкой»

А дальше на протяжении 1994–1996 годов плавно сменились версии 3.1 и 3.2, не прославившиеся какими-либо громкими новшествами. После чего в мае 1997 года выходит версия 3.3, ознаменовавшаяся появлением в ней XFree86 Acceleration Architecture (XAA). Архитектура эта, как явствует из названия, обеспечивала ускоренный вывод графики – пока еще двумерной: как это ни трудно нынче представить, но тогда последняя еще нуждалась в акселерации. Именно на ее почве расцвели пышным цветом те самые X-сервера, привязанные к видеочипам, о которых шла речь в заключительной части прошлой статьи.

За два с половиной года активного развития версии 3.3 она пережила более полудюжины минорных релизов, характеризовавшихся, однако, весьма мажорными нововведениями. Описывать их последовательно было бы скучно, да нынче и проследить их поверсионно сейчас уже нелегко. Поэтому остановлюсь на тех из них, которые сыграли важную роль в постановке под названием «десктопный Linux».

О постоянном совершенствовании X-серверов, обеспечивавших базу для всего остального, я уже говорил. Далее, важным оказалось развитие поддержки шрифтов. Первоначально в XFree86 применялись, во-первых, растровые шрифты (форматов bdf и pcf), во-вторых, векторные шрифты формата PostScript (ATM Type 1). Сфера применения первых была ограничена – они использовались преимущественно в терминальных программах и текстовых редакторах. Назначение же вторых, теоретически, вполне универсально. Практически же их широкому использованию мешало. во-первых, низкое качество самих шрифтов, во-вторых – не лучший их рендеринг.

› **XFree86 в Mandrake/RE 7.0 учатся русскому.**

Легенда о двух Shift'ах

Во времена самостийных русификаций DOS и ее приложений переключатели латинской и кириллической раскладок бывали самыми причудливыми. Однако, когда русский язык начал официально поддерживаться этой ОС (начиная с версии 4.01), в этом качестве утвердилась комбинация левого и правого Shift'ов. И ей суждена была долгая жизнь – но благодаря не DOS'у, а Linux'у.

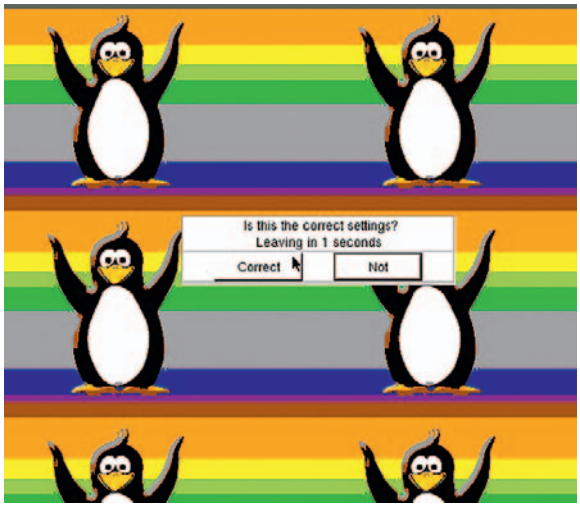
В первых версиях Red Hat, самостоятельно, без участия российских пользователей, поддерживавших русский язык, выбор его на стадии установки автоматически приводил к тому, что умолчальной становилась кириллическая раскладка (причем об этом нигде не говорилось). Что на стадии создания аккаунтов делало невозможным ввод корректных паролей администратора и пользователя, если не переключиться на латиницу. Переключение это программе установки (и только в ней) осуществлялось комбинацией обоих Shift'ов – но это нигде не было документировано. И не одно поколение пользователей Red Hat, а затем и Fedora, имело повод поупражняться в солдатской смекалке, угадывая сначала, почему в ответ на ввод пароля выдается сообщение об ошибке, а потом – о способе ее устранения. Ибо последние рецидивы данного бага (давно превратившегося в фичу) я наблюдал в Fedora 11.

До решения второй проблемы было еще далеко. А вот в отношении первой в описываемое время произошли определенные подвижки, выразившиеся в появлении поддержки шрифтов True Type. Что сначала дало возможность использовать шрифты от Microsoft, штатно применявшиеся в Windows, начиная с версии 3.1 (Arial, Times New Roman, Courier New и так далее), по чьему-то недосмотру оказавшиеся в почти свободном доступе. А потом оказалось толчком для разработки качественных ttf-шрифтов, уже полностью свободных (Bitstream Vera, DejaVu, Liberation).

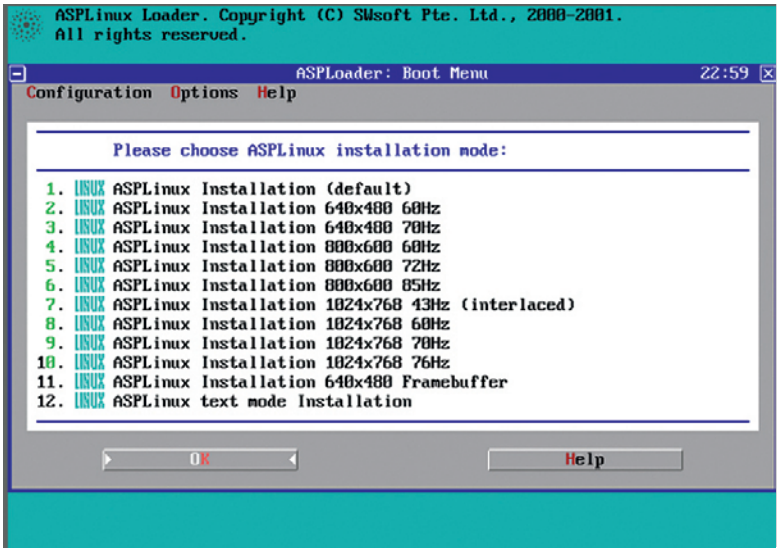
Впрочем, тема шрифтов в Иксах заслуживает отдельного рассмотрения. А сейчас надо сказать пару слов о поддержке устройств ввода – то есть клавиатур и мышей.

О самих по себе клавиатурах говорить нечего – как устройства они поддерживались всеми вариантами Иксов испокон веку. А вот с раскладками для различных языков (точнее, наборов символов) связана забавная история, касающаяся, разумеется, нашей любимой кириллицы.

Ввод кириллицы в кодировках ISO 8859-5 (по историческим причинам изначально принятый в русских вариантах проприетарных UNIX'ов, в частности, в SunOS) и KOI8-R (утвердившейся, благодаря Андрею Чернову, именуемому также ache, во всех свободных UNIX-подобных системах), поддерживался еще в XFree86 2.X, однако методом, который считался идеологически неправильным. Полноценная поддержка кириллических раскладок стала



› Mandrake/RE 7.0: тест видеорежима.



› Первый ASPLinux: так выбиралось разрешение.

возможной только в версиях 3-й ветки – правда, ценой несовместимости с более старыми русифицированными приложениями, например, *Netscape Navigator* (легендарный предшественник современных *FireFox*'а и *Thunderbird*'а, безвременно почивший под натиском тогда еще юного хищника Microsoft).

Параллельное использование более чем одной раскладки клавиатуры (впрочем, в те времена их число ограничивалось двумя) требовало средства переключения между ними. И такое средство еще в версиях 2.X было жестко встроено в саму раскладку. В частности, переключение с латиницы на кириллицу и обратно было привязано к клавише CapsLock (а функцию перевода алфавитных клавиш в верхний регистр выполняла комбинация Shift+CapsLock).

В XFree86 3.X появилась возможность переназначения переключателя раскладок, чем народ немедленно воспользовался: для переключения с латиницы на кириллицу пошли в ход Windows-подобные комбинации типа Control+Shift и Alt+Shift, и даже сакраментальные «два шифта».

При этом поначалу наличие «заказного» переключателя раскладок не исключало и использование переключателя встроенного, то есть все той же клавиши CapsLock. Что, казалось бы, никому не мешало – как и наличие бочек с красной и черной икрой при входе в магазин Рабиновича на Дерибасовской улице. Однако было сочтено излишеством – и в версии XFree86 3.4 встроенный переключатель раскладок исчез. Что скоро, с появлением средств автоматического конфигурирования Иксов, доставило много дополнительных развлечений не одному поколению начинающих юзеров – линуксоидов и берклианцев.

Однако прежде чем отсутствие встроенного переключателя раскладок проявило себя во всей красе, произошла та самая забавная история, о которой я недавно упоминал. Суть ее была в том, что в один прекрасный момент, с выходом XFree86 версии 3.3.2, русская раскладка для кодировки KOI8-R без всяких видимых причин просто перестала работать. Точнее, она делала вид, что работала, но при выводе получалась абракадабра из русских букв.

Дело оказалось в том, что раскладка для кодировки KOI8-R подменялась таблицей для ISO 8859-5. А причина заключалась в банальной опечатке в исходниках главной Иксовой библиотеки – *xlib*. Более ни на что эта опечатка не влияла, и потому ее выявление заняло довольно много времени. Это было сделано Иваном Паскалем, который и написал соответствующий патч для версии 3.3.2, штатно включенный в следующий релиз (3.3.3).

Тем не менее, XFree86 в рамках ветки 3.3 от версии к версии все совершенствовалась, пока система не приобрела законченный вид в релизе 3.3.6, вышедшем в декабре 1999 года. Она тут же была включена во все дистрибутивы Linux, и использовалась на протяжении долгого времени.

> «Переключатель раскладок доставил много развлечений юзерам.»

Однако на этом развитие 3-й ветки XFree86 фактически прекратилось – уже в марте 2000 года появляется первый релиз ветки 4-й, кардинально переработанной. Правда, разработчики XFree86 совершенно четко позиционировали версию 4.0 как экспериментальную, и майнтайнеры всех распространенных дистрибутивов вняли их предупреждению, включая ее в состав своих сборок в качестве опции: как основная графическая система, еще долгое время применялась версия 3.3.6.

»

Настройка XFree86: как это делалось встарь

Говорят, что во времена совсем былинные настроить Иксы можно было только лобовым созданием их конфигурационного файла в текстовом редакторе. Однако затем появились механизированные средства для решения этой задачи, и первое из них носило имя *xf86config*, издревле входившее в штатную поставку XFree86. Оно запускалось из командной строки и работало в интерактивном режиме. То есть – требовало нудного ответа на множество вопросов. И любая ошибка могла быть исправлена только одним способом: выходом из конфигуратора и повторением процедуры с самого начала. Однако это средство обладало одним несомненным достоинством – при должной аккуратности позволяло добиться результата (в виде сгенерированного файла **/etc/XFree86.conf**), почти не требующего ручной доводки (за исключением некоторой докрутки кириллицы и, для некоторых CRT-мониторов, юстирования частотных характеристик развертки).

Настройка XFree86: как это делалось в Одессе

В XFree86 появились меню-ориентированные программы настройки: *Xconfigurator* с псевдографическим интерфейсом; *XF86Setup*, работавшая в графическом режиме VGA; и *xf86cfg*, допускающая как текстовый, так и графический варианты запуска, причем в последнем она пыталась максимально точно определить параметры видеоподсистемы. Разумеется, работать через меню, да еще и (в графических вариантах) управляемом мышью, было куда проще, чем отвечать на кучу вопросов. Да и вернуться к ошибочно определенному пункту было куда легче, чем перезапускать заново, с нуля, *xf86config*. Однако объем ручной правки после работы любой из перечисленных утилит был куда больше, чем при использовании последнего.

Хроника поступательного движения

Главным новшеством XFree86 4-й ветки была ликвидация зоопарка X-серверов, расплодившихся по принципу «один видеочип – один сервер». Отныне герой (то есть X-сервер) остался один. А вся чипо-специфическая его часть выносится в отдельные модули (по старой традиции называемые драйверами).

Далее, в XFree86 4.X появляются собственные средства автоконфигурирования. Если раньше к таковым понятие «авто» можно было применить чисто условно (см. врезки о настройке), то теперь запуск команды

```
# X -configure
```

давал неизменно превосходный результат в виде конфигурационного файла **/etc/X11/XF86Config** – компактного, но при этом исчерпывающе прокомментированного. А самое главное – почти всегда работающего без всяких дополнительных телодвижений. Лишь в нашем кириллическом окружении он требовал некоторой минимальной ручной доводки.

Разумеется, сказанным усовершенствования «четверки» не исчерпывались: тут были и возможность работы с двумя мониторам, и автоматическое распознавание большинства моделей «колесатых» мышей (ранее для их поддержки требовался отдельный пакет *imwheel*), и способность работать с графическими планшетами «рисовального» назначения.

Все эти «улучшательства», постепенно накапливавшиеся в ходе коротких жизненных циклов минорных релизов 4.0.X, привели к тому, что релиз 4.1, вышедший в июне 2001 года, широко распространился в дистрибутивах Linux в качестве «Иксов по умолчанию». А с появлением в начале 2002 года релиза 4.2 «тройка» из их состава практически исчезла, уцелев разве что в ультраконсервативных системах типа Debian'а.

На этом поступательное развитие XFree86 не остановилось: в сентябре 2002 года выходит корректирующий релиз 4.2.1, а за ним, в феврале 2003, «полумажорный» 4.3. В рамках последнего продолжают накапливаться усовершенствования, призванные воплотиться в будущем релизе 4.4, уже совсем «мажорном». Однако тут происходит поворот судьбы, значение которого для дальнейшего развития и Иксов, и Linux'а трудно преувеличить – хотя и до сих пор не ясно, как оценить.

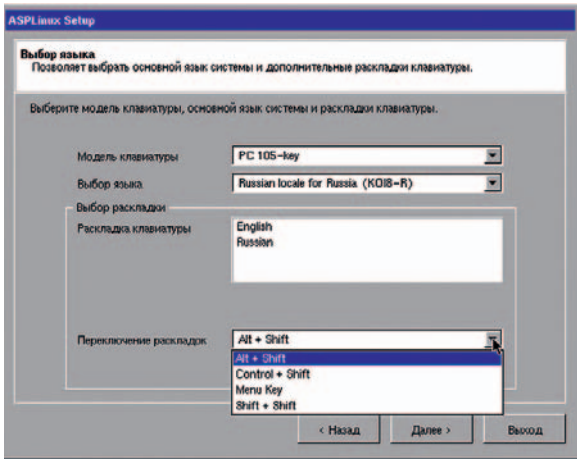
Предпосылки Великого раскола

Все указанные усовершенствования, хотя и существенно улучшали жизнь пользователей, но никак не могли рассматриваться в качестве революционных. Что не нравилось некоторым участникам проекта XFree86, в первую очередь – одному из его основоположников, Кейту Паккарду [Keith Packard]. И о дальнейших событиях существуют противоречивые свидетельства.

Основная часть разработчиков проекта утверждала тогда, что Кейт втихаря начал разработку проекта параллельного. И в результате в марте 2003 года его исключают из «партии XFree86» с закрытием доступа к «партийному распределителю» – дереву исходников. Сам же Паккард отрицал партизанский характер своих действий, хотя и активно критиковал бывших коллег за медлительность в рассмотрении патчей от сторонних разработчиков и общий консерватизм. И, действительно, после изгнания «из рядов» создал собственный форк Иксов – тот самый, которому суждено было стать современным проектом X.org. Обещая, что он будет развиваться более динамично и «инновационно».

> «Поводом для раскола послужили лицензионные противоречия.»

Насколько эти обещания были выполнены на первом, еще «безымянном», этапе развития нового проекта, судить трудно. Потому что это не был еще сам Великий раскол: в итоге основу форка X.org составил один из пре-релизов XFree86 – 4.4 RC2. А поводом для собственно раскола послужили лицензионные противоречия. Но это – совсем другая история, которая принадлежит скорее политике. LXF



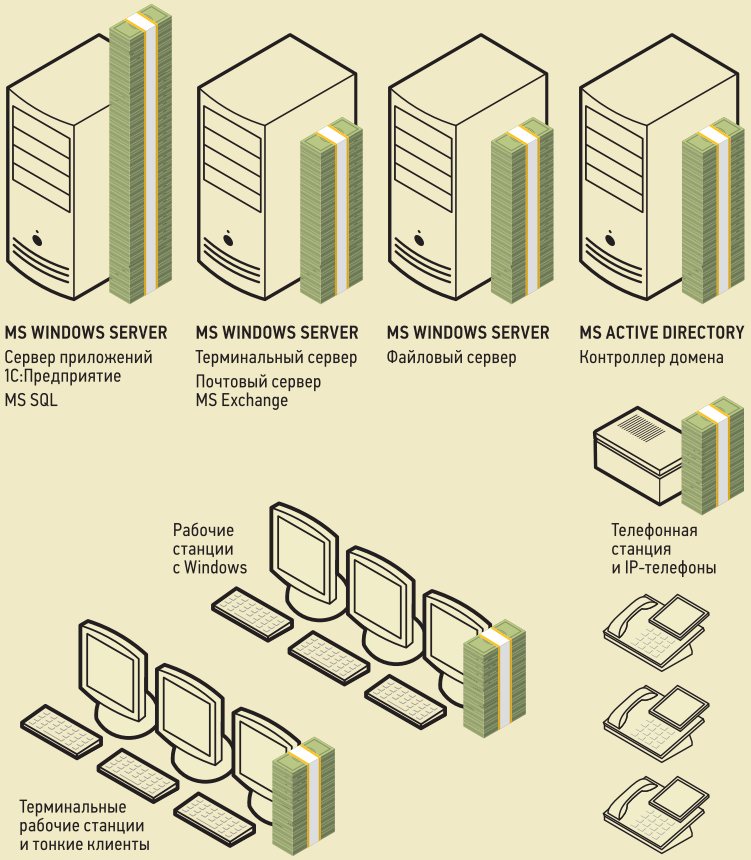
› Клавиатура в ASPLinux: тот самый переключатель раскладок.

Иксы в дистрибутивах

В конце 90-х годов ряд дистрибутивов, проявлявших наибольшую любовь к пользователям, штатно включили средства настройки Иксов в свои инсталляторы. В числе первых на этом поприще отметились Mandrake и Caldera OpenLinux – приоритет сейчас уже не определить. Да он и не очень важен. Более существенно то, что средства настройки Mandrake были унаследованы сначала его Russian Edition, а потом AltLinux'ом. Где были расширены, углублены и закалены в боях с кириллицей (или, правильней сказать, за кириллицу в Иксах). А под идейным влиянием дистрибутива от Caldera развивался инсталлятор *ASPLinux*, выпускавшийся одноименной фирмой. А поскольку фирма имела соплеменное нам происхождение, то и он доблестно показал себя на ниве кириллизации Иксов.

ВИРТУАЛЬНАЯ ИТ-ИНФРАСТРУКТУРА ПРЕДПРИЯТИЯ НА БАЗЕ СВОБОДНОГО ПО

На базе Windows*



На базе GNU/Linux и виртуализации*

GNU/LINUX

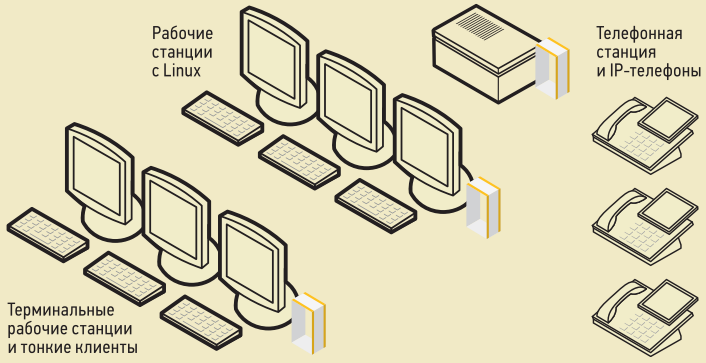
Терминальный сервер
Клиент 1С:Предприятие под WINE@Etersoft
IP-телефония Asterisk
Почтовый сервер Zimbra
Хранилище документов Alfresko

Терминальный сервер
PostgreSQL
Сервер приложений 1С:Предприятие

Файловый сервер

MDS Контроллер домена
DHCP, OpenLDAP, WebConfig

Почтовый сервер, файловый сервер, контроллер домена, базы данных SQL, 1С:Предприятие, IP-телефония – **и все это на одном физическом сервере в виртуальных машинах!**



ПРЕИМУЩЕСТВА ТЕРМИНАЛЬНОГО РЕШЕНИЯ НА БАЗЕ GNU/LINUX:

- экономия средств на лицензировании ПО
- сокращение затрат на оборудование за счет виртуализации
- единая инфраструктура на базе свободного ПО
- защита от вирусов и других угроз безопасности

Позвоните сейчас!
Назначьте встречу с нашим специалистом!

Офис в Санкт-Петербурге
Лиговский пр., 50, корпус 15
Тел.: (812) 309-06-86,
факс: (812) 640-49-90

Офис в Москве
Красноказарменная ул., 17
(в помещении АТС МЭИ)
Тел.: (499) 271-49-54

Linux center
www.linuxcenter.ru

* Сравнительная стоимость программного обеспечения в комплексном решении для малого и среднего бизнеса

Свободное ПО + виртуализация = экономия в 10 раз!

Лицензионные платежи отсутствуют, планируется бюджет только на внедрение и техническую поддержку

Что за штука...

WebRTC

Марко Фиоретти изучает WebRTC – проект, который обещает сделать связь в реальном времени доступной через браузеры и другие web-приложения.

В Какие возможности предоставляет этот WebRTC?

О Самый простой, хоть и обтекаемый ответ – видеозвонки в реальном времени и обмен данными через браузеры.

В Но ведь это уже возможно в Сети. Мне известны такие сайты.

О Строго говоря, вы правы. Но на практике это возможно только между двумя пользователями и при условии совпадения браузера, ОС, внешних плагинов, знака Зодиака, группы крови и прочих недокументированных требований – также меняющихся в зависимости от сайта. В то время как с WebRTC максимальное количество кода, необходимого для осуществления коммуникации, стандартизовано и собрано вместе, чтобы можно было его интегрировать в любой современный сетевой обозреватель.

В С тем, чтобы у разработчиков приложений и web-дизайнеров было меньше хлопот, верно?

О Именно! Цель – создание открытых библиотек, которые позволят несколькими строчками JavaScript выразить что-то вроде «отправить вывод локальной web-камеры в окно удаленного браузера». И быть уверенным, что все будет работать вне зависимости от того, какой именно там браузер. Можете представить, сколько web-сервисов смогут это использовать?

В Много. Пожалуй, так много, что конкретно и не сообразить. Пример не приведете?

О Ограничимся семейными потребностями? Ну, вообразите онлайн-игры, обучение, интернет-магазины и сервисы вроде «щелкните здесь и покажите нам, как бы вы хотели изменить дизайн своей куртки».

В Ну ясно. Даже социальные сети могут применять WebRTC, да?

О Конечно. Сделать видеозвонок одним щелчком по контакту будет быстрее и проще, чем сейчас, в любой социальной сети. Или между разными социальными сетями.

В Похоже, WebRTC составит конкуренцию и профессиональным приложениям для видеоконференций.

О И да, и нет. Конечно, Web-RTC годится и для этого, и наверняка кто-то его применит. Но похоже, что прямой конкуренции не будет. Профессиональные продукты для сетевых видеоконференций имеют свою специфику и целевую аудиторию. Суть WebRTC в унификации коммуникаций в реальном времени. Как именно – это другой вопрос, по крайней мере, в средне/долгосрочной перспективе.

В Здорово. Значит, я смогу подключаться к сайтам WebRTC через Skype?

О Поговаривали, что Skype планирует создать клиент на основе браузера, способный подключаться напрямую к клиентам Skype. Такой клиент может использовать и WebRTC. Но напомню, что смысл WebRTC – в упразднении автономных и несовместимых программ наподобие Skype, через перевод их функций в браузеры.

В Понятно. Но сможет ли браузер с WebRTC позвонить на обычный телефон?

О Такие инструменты уже существуют – например, webrtc2sip (http//code.google.com/p/webrtc2sip), программа, выполняющая функции шлюза между соответствующе укомплектованными браузерами и традиционными телефонными сетями. Будут ли и насколько будут такие инструменты масштабироваться и применяться, пока не известно.

В Давайте заглянем под капот. Каковы основные компоненты WebRTC?

О Это три интерфейса программирования приложений: MediaStream, RTCPeerConnection и RTCDataChannel.

В Чем занимается, например, первый из них, MediaStream?

О MediaStream предоставляет доступ (с разрешения пользователя) к таким потокам данных, как выводы web-камеры и микрофона. Благодаря MediaStream, приложения на базе браузера могут также управлять этими потоками в реальном времени, например, добавлять спецэффекты.

В Это касается исключительно аудио и видео?

О Нет. Согласно спецификации, в финале MediaStream должен включать и поддерживать любые потоки в реальном времени, в том числе данные сенсоров и подключаемых устройств.

В Прямо технократия!.. Ну, а что у нас с RTC PeerConnection?

О Эта часть WebRTC позволит браузерам делать нечто трудновыполнимое сейчас: напрямую обмениваться медиа-данными в реальном времени по каналам, являющимся одновременно зашифрованными и динамически оптимизированными, чтобы поддерживать наилучшее качество с доступным расширением.

В А каким образом мне может пригодиться Data Channel?

О Рано или поздно, общаясь с друзьями в вашем браузере, вы захотите обмениваться с ними файлами, возможно, просто перетаскивая иконки. Как раз Data Channel и добавит такую возможность в WebRTC-общение. Однако по состоянию на январь 2013 его внедрение все еще под большим вопросом.

В Так ведь браузеры зачастую защищены межсетевыми экранами, либо не имеют публичных IP-адресов. Будет ли при этом работать WebRTC?

О Если ваш босс велит системному администратору компании заблокировать все интернет-соединения, то ничего не поделаешь. В других, менее радикальных условиях, WebRTC должен работать без проблем, поскольку в нем есть соответствующие технологии.

> «Смысл WebRTC — в упразднении автономных программ наподобие Skype.»

В Поясните, пожалуйста, как эти технологии работают.

О Через серверы, помогающие, к примеру, одноранговым узлам WebRTC, упрятанным за NAT, определять свои публичные IP. Если вам непременно нужны все эти гнусные, маловразумительные технические подробности, обратитесь к разделам сайта, посвященным протоколам TURN, STUN и ICE.

В WebRTC, надеюсь, является открытым проектом?

О Да, предложение W3C по стандартам вы найдете на www.w3.org/TR/webrtc. Код представлен в виде сборки библиотек JavaScript, которые используют, или будут использовать, несколько API.

В Есть ли у WebRTC за спиной какие-нибудь известные компании-разработчики?

О Ключевые фигуры – Mozilla, Opera и Google: им эта технология может пригодиться для пользователей Talk и Hangout.

В А как насчет Apple? Или Microsoft, который, вдобавок, владеет Skype?

О Apple, похоже, пока выбрали позицию «поживем – увидим». Microsoft и Skype подключились к официальной дискуссии позднее прочих и предложили собственный подход.

В И в чем этот собственный подход заключается?

О Предложение Microsoft носит название Customizable, Ubiquitous Real-Time Communication [Настраиваемые повсеместные коммуникации в реальном времени], или коротко CU-RTC. Предполагается, что этот проект будет менее амбициозен и более прост в реализации, нежели WebRTC, однако зависимость от браузера будет такой же, как сейчас, а может, и увеличится. Mozilla и их соратники, естественно, утверждают, что и с WebRTC все не так уж сложно, а ограничения если и наличествуют, то временные. Поживем –увидим.

В Станет ли WebRTC глобальным решением для общения в реальном времени, или что-то существенное ему не охватить?

О Ну, WebRTC представляет собой набор спецификаций, касающихся только технических вопросов передачи голоса, видео и данных между приложениями в реальном времени. И если бы возникло желание или потребность перевести все наши голосовые и видеозвонки на WebRTC и браузер, было бы две проблемы: первая касается формальной стандартизации, и ею ни в коем случае нельзя пренебрегать, дабы защитить пользователей от несовместимостей версий.

А если бы общение через браузер должно было заменить нынешние стационарные или мобильные телефонные звонки, понадобилось бы создать всю остальную инфраструктуру, от прямых экстренных вызовов до списков абонентов и соглашений на уровне услуг. Но ведь пока такого желания и такой потребности нет, верно?

В Верно. Но мне чрезвычайно любопытно. Могу я уже сегодня попробовать WebRTC?

О Конечно, можете – на последней версии *Chrome* (или, приложив малость усилий, на *Internet Explorer*). Заходите на http://www.webrtc.org/demo, убедитесь, что ваша web-камера, микрофон и колонки подключены и попробуйте одну из предложенных там демок.

На устройствах Android или iOS можете попробовать обозреватель *Bowser* (https://labs.ericsson.com/apps/bowser), но поскольку в нем пока используется старый протокол сигнализации, вы не сможете связаться с пользователями *Chrome*.

В Ну, а что будет со звонками на обычные телефоны?

О Разработчики открытого приложения *sipml5* (http://sipml5.org) говорят, что с его помощью можно прямо из браузера получать доступ как к сетям мгновенных сообщений, так и к SIP-шлюзам, которые способны переадресовать ваши звонки в простые телефонные сети.

В После всего этого я хочу знать больше о WebRTC!

О За более полной информацией о том, как устроен WebRTC, обратитесь к весьма объемному, но доходчивому учебнику на www.html5rocks.com/en/tutorials/webrtc/basics. Чтобы быть в курсе последних новостей, следите за официальным блогом WebRTC (www.webrtc.org/blog) или @webrtc в Twitter. LXF

По рецептам доктора Брауна

Д-р Крис Браун
Доктор обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

Эзотерическое системное администрирование из причудливых заворотов кишок серверной

Учимся править

Я провожу много времени в учебных классах, глядя на то, как разные люди управляют компьютерами. Забавно бывает наблюдать широкий диапазон подходов.

Изредка мне достается студентка, умеющая печатать слепым методом. Строки текста у нее (а это почему-то всегда она) без усилий перепархивают с клавиатуры на экран, и смотреть на это приятно.

На противоположной стороне спектра находятся хронические клавофобы – люди, которые пускаются на возможные уловки, чтобы ничего не набирать.

Например, им нужно набрать имя файла **/etc/fstab**. Они прокручивают назад несколько страниц текста (разумеется, мышью), чтобы найти это имя, скопировать и вставить его. Они совершенно не задумываются о том, что таким способом у них уходит вшестеро больше времени.

Отвращение к клавиатуре

Лишь немногим меньше раздражают те, кто потратил всего три минуты на изучение *Vi* и все время остается в режиме вставки и перемещает курсор стрелками, как тот, кто купил «Феррари», но не хочет возиться с рычагом переключения передач.

Затем есть такие, кто набирает команду, откидывается на спинку стула и восхищается командой, прежде чем нажать Enter, как будто команда должна созреть, чтобы достичь полного эффекта.

Иногда учебные задания требуют от студентов работать в парах, и очень любопытно наблюдать за стратегиями, которые те вырабатывают для объединения усилий во время лабораторной работы. Одна примечательная парочка решила, что один будет управлять клавиатурой, а другой – мышью.

Тот факт, что у них вроде бы все получилось, продемонстрировал удивительную и редкостную гармонию мышления.

chris.linuxformat@gmail.com

ImageMagick

Рассмотрим утилиту, которая делает с изображениями все, что ни пожелаешь, разве что не рисует их.

Так как мой раздел посвящен системному администрированию, то, решив привлечь ваше внимание к удивительному набору утилит управления изображениями под названием *ImageMagick*, я немного сойду с лыжни. Идея о наборе утилит для редактирования изображений, работающих чисто в командной строке, может удивить даже самых опытных любителей Linux, но *ImageMagick* именно таков (хотя для некоторых операций есть оболочка в виде меню).

От широты возможностей *ImageMagick* захватывает дух. Хотя им нельзя закрасить на фото усы вашей тещи, почти со всем прочим он справится. Можно преобразовать изображение в другой формат (поддерживается более 100), перевернуть, отразить, масштабировать, обрезать и трансформировать изображение, размыть его и увеличить резкость, изменить цвета, определить края, добавить текст и фигуры и многое другое.

Формат от скрипта

Одно из преимуществ командной строки, разумеется, в построении из команд скриптов. Например, вот цикл в скрипте, который преобразует все файлы JPG в текущем каталоге в PNG:

```
$ for f in *.jpg; do convert $f $(basename $f.jpg).png; done
```

При поддержке практически всех известных человечеству форматов возможности программы неисчерпаемы.

Охватить все возможности ImageMagick всего в нескольких примерах нельзя, но вот два. Первый создает изображение, содержащее текст, который располагается вдоль 360-градусной дуги:

```
$ convert -background khaki -fill sienna -font acmesa.TTF -pointsize 144 label:"Рецепты доктора Брауна. " -distort Arc 360 arctext.png
```

Мой второй пример создает коллаж из миниатюр всех изображений, которые есть в каталоге Pictures:

```
$ montage -label '%t\n%[width]x%[height]' -size 512x512 '/home/chris/Pictures/*.jpg[120x90]' -auto-orient -geometry +5+5 -tile 5x -frame 5 -shadow photos.png
```

Результат – аккуратная серия миниатюр, каждая с заголовком и в красивой рамке с тенью. Эти примеры иллюстрируют работу только двух команд – **convert** и **montage**. Есть еще девять: **animate**, **compare**, **composite**, **conjure**, **display**, **identify**, **import**, **mogrify** и **stream**. В совокупности у них столько опций и параметров, что ошалеет самый отпетый наркоман командной строки. К счастью, на www.imagemagick.org очень много готовых примеров, покрывающих все: от размытия до аффинных преобразований.

Наряду с управлением *ImageMagick* из командной строки (или из скрипта), есть библиотеки, предоставляющие доступ ко всему этому великолепию для работы с изображениями на языках C, Java, Perl, PHP и Ruby. *ImageMagick* почти наверняка есть в вашем дистрибутиве, и его установка должна быть простой. Существует версия также и для Windows.

Российский Интернет Форум

РИФ+КИБ 2013

#RIF2013 ПРИБЫВАЕТ ПО РАСПИСАНИЮ!

17-19 АПРЕЛЯ | ПАНСИОНАТ «ПОЛЯНЫ»

WWW.RIF.RU

Стек LAMP

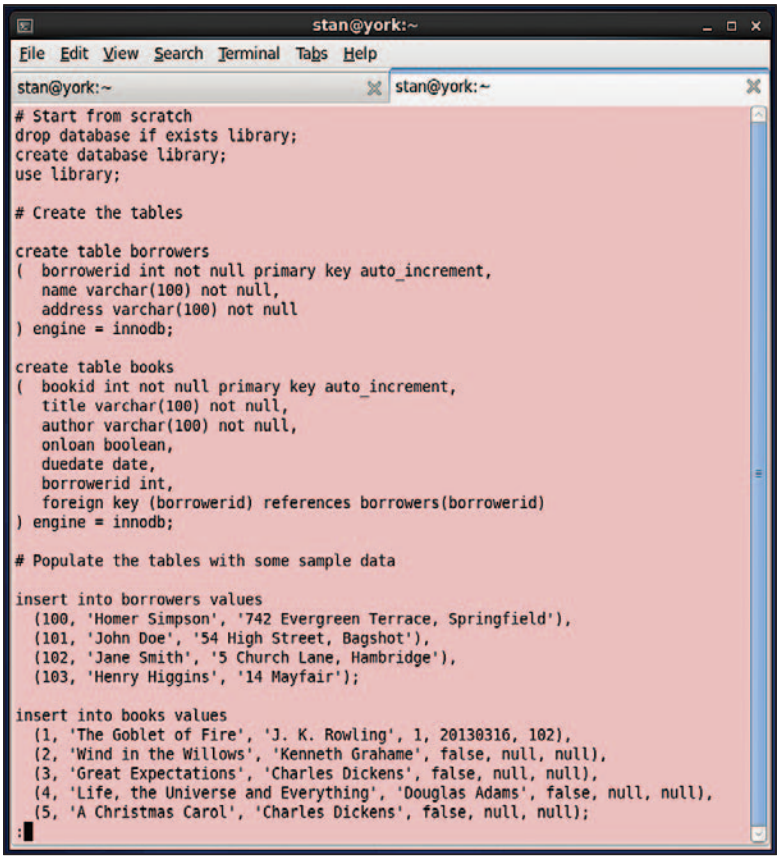
M – это *MySQL*. Третья часть нашей серии, и в ней мы осваиваем базы данных.

В этом месяце мы рассмотрим «M» в стеке LAMP – *MySQL*. Сперва вкратце ознакомимся с теорией баз данных, затем перейдем к SQL и, наконец, установим *MySQL* и создадим базу данных. Как и в предыдущих статьях, здесь я пользуюсь CentOS 6.2 – и советую вам последовать моему примеру и установить его на виртуальную машину.

Пособие по нырку в базы данных

На то, чтобы стать гуру в базах данных, можно потратить всю жизнь, но основы не слишком сложны. Начнем с начала: в базах данных информация хранится в таблицах, и у каждой таблицы есть имя. Таблица состоит из столбцов и строк. У каждого столбца есть имя и тип данных. Эти типы данных отчасти зависят от используемого сервера баз данных, и необязательно должны напрямую соответствовать типам данных, используемым в серверных языках программирования. Например, *MySQL* поддерживает 8-битные целые числа в виде типа TINYINT, которого нет в PHP.

Один столбец внутри каждой таблицы назначается первичным ключом. Он должен быть уникален внутри таблицы; иными словами, запросив данные из таблицы для заданного первичного ключа, вы получите в ответ максимум одну строку. Так же часто один или несколько столбцов содержат так называемые внешние ключи, которые представляют собой перекрестные ссылки на первичные ключи в других таблицах. Структура базы данных – ее таблицы, их столбцы и типы данных и взаимосвязи между ключами – называется схемой базы данных.




```
# Start from scratch
drop database if exists library;
create database library;
use library;

# Create the tables

create table borrowers
(  borrowerid int not null primary key auto_increment,
   name varchar(100) not null,
   address varchar(100) not null
) engine = innodb;

create table books
(  bookid int not null primary key auto_increment,
   title varchar(100) not null,
   author varchar(100) not null,
   onloan boolean,
   duedate date,
   borrowerid int,
   foreign key (borrowerid) references borrowers(borrowerid)
) engine = innodb;

# Populate the tables with some sample data

insert into borrowers values
  (100, 'Homer Simpson', '742 Evergreen Terrace, Springfield'),
  (101, 'John Doe', '54 High Street, Bagshot'),
  (102, 'Jane Smith', '5 Church Lane, Hambridge'),
  (103, 'Henry Higgins', '14 Mayfair');

insert into books values
  (1, 'The Goblet of Fire', 'J. K. Rowling', 1, 20130316, 102),
  (2, 'Wind in the Willows', 'Kenneth Grahame', false, null, null),
  (3, 'Great Expectations', 'Charles Dickens', false, null, null),
  (4, 'Life, the Universe and Everything', 'Douglas Adams', false, null, null),
  (5, 'A Christmas Carol', 'Charles Dickens', false, null, null);
:
```

› **Это скрипт для создания и заполнения данными базы данных библиотеки.**

Рассмотрим пример. На рисунке показаны две таблицы books [книги] и borrowers [читатели], которые могут составить базу данных, используемую библиотекой, выдающей книги на дом. База данных позволяет нам следить за книгами в библиотеке – выданы они или нет, и если да – кто их взял и когда вернет.

В таблице book каждой книге назначен уникальный идентификатор книги; он выступает в роли первичного ключа таблицы. Может возникнуть соблазн использовать как первичный ключ ISBN книги – в конце концов, он обязан быть уникальным. Но тогда в библиотеке нельзя будет держать больше одного экземпляра каждой книги. Использование собственного «идентификатора книги» в качестве первичного ключа позволяет избежать этой проблемы. Посмотрев на таблицу на рисунке, мы видим, что одну из книг взял читатель 102. Это внешний ключ. 102 ссылается на первичный ключ в таблице borrower. Посмотрев в эту таблицу, мы видим, что книгу взяла Джейн Смит [Jane Smith].

Язык структурных запросов

Для создания таблиц, выполнения к ним запросов и их обновления используется древний язык под названием SQL (Structured Query Language – язык структурированных запросов). Он появился в 1974 году и стал стандартом ANSI в 1986, и этим сильно отличается от всех прочих языков программирования, с которыми вы можете быть знакомы. О нем написаны целые книги, но так как в этом месяце мне не выделили всю площадь журнала, придется довольствоваться несколькими примерами, чтобы вы поняли основную идею.

Для начала, вот пара SQL-запросов:

```
select * from books where author = 'Charles Dickens'
select author,title from books where onloan = 1
```

Каждый из этих запросов возвращает «результирующее множество» – набор из нуля или более строк, содержащий соответствующие запросу записи. В первом примере * означает, что мы хотим получить все столбцы. Посмотреть в таблицу, вы увидите, что мы получим две строки (так как Чарльз Диккенс написал две из наших книг). Во втором примере в результатах нам нужны только столбцы author и title, и так как только одна книга из таблицы выдана на руки, результат будет состоять из одной строки.

Не все запросы возвращают строки данных. Следующий запрос возвратит просто количество выданных книг:

```
select count(*) from books where onloan = 1
```

Теперь предположим, что в библиотеку поступила новая книга. Вот SQL-запрос для добавления новой строки в таблицу:

Немного личного

В годы моего становления как программиста я занимался т. н. «научными» вычислениями в различных университетских исследовательских группах. Я мог написать программу, выполняющую двумерное преобразование Фурье или моделирующую физику пузырьковой камеры (клянусь!), но вручи мне кто базу данных – и я не знал бы, что с ней делать. Итак, я пришел к реляционным данным довольно поздно и не провел жизнь, варясь в ссылочной целостности данных и их первых нормальных форм. Но почти все, кого я встречаю в учебных классах, считают базы данных сердцем своих приложений.

```
insert into books values (6, 'Birdsong', 'Sebastian Faulks', 0, null, null)
```

Обратите внимание, что первоначально книга никому не выдана. Ключевое слово **null** означает, что элементу не присвоено никакого значения.

И вот наша библиотека решила избавиться от единственной книги Дугласа Адамса [Douglas Adams] (позор!), и нам нужно удалить ее из базы данных следующим SQL-запросом:

```
delete from books where bookid = 4
```

Обратите внимание, что поскольку здесь мы выбираем данные по первичному ключу, мы удалим максимум одну строку. Другой пример – если бы мы захотели удалить все книги Чарльза Диккенса [Charles Dickens] (еще худший позор!), это можно было бы сделать так:

```
delete from books where author = 'Charles Dickens'
```

Наконец, представим, что читатель Джон Доу [John Doe] сообщил библиотеке об изменении своего адреса, поэтому нужно обновить базу данных следующим образом:

```
update borrowers set address = '15 Lemon Street, Liverpool' where name = 'John Doe'
```

В этих примерах есть несколько вещей, на которые нужно обратить внимание. Во-первых, ключевые слова SQL (вроде **select**, **where**, **insert** и **delete**) я писал в нижнем регистре. Это нормально, так как SQL не чувствителен к регистру; но большинство рьяных сторонников SQL написали бы их заглавными буквами. Во-вторых, здесь мы только смотрим на выражения языка SQL. Далее на нашем уроке мы увидим, как выполнять эти команды в клиенте командной строки *MySQL*, а через месяц – из кода PHP, но пока это чистый SQL.

Более сложные запросы

Прежде чем завершить наше мини-руководство по SQL, мне хочется показать вам чуть более хитрый запрос. Пусть нам нужно вывести список заголовков книг и читателей для книг, выданных на руки. Эта информация разбита между двумя таблицами, поэтому нам нужно объединить результаты из обеих. Запрос выглядит следующим образом:

```
select books.title,borrowers.name from books,borrowers where books.onloan = 1 and books.borrowerid = borrowers.borrowerid ;
```

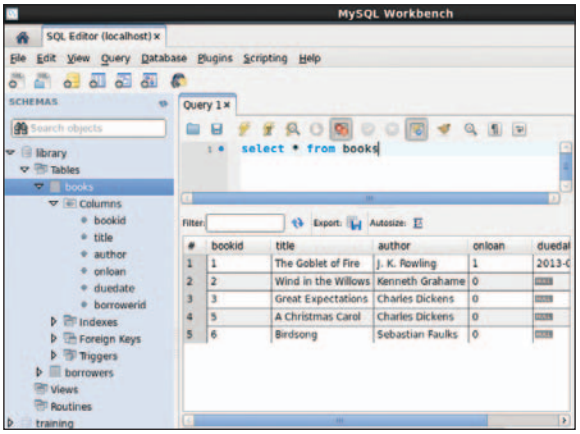
Обратите внимание, что теперь имена столбцов нужно указывать с именами таблиц, например, **books.title** вместо просто **title**. Такой типа запроса называется объединением. А в выражении **where** – два условия, соединенные логическим «и (**and**)». Второе условие – **books.borrowerid = borrowers.borrowerid** – гарантирует, что мы получим только те сочетания данных из двух таблиц, которые соответствуют одному и тому же взятию книги на руки.

Здесь мы коснулись SQL лишь в общих чертах. Мы не говорили о командах создания новых баз данных и таблиц или о хранимых процедурах – которые по сути являются именованными фрагментами кода SQL, хранимыми на сервере. Мы также вовсе не упомянули об управлении доступом. В *MySQL* есть довольно тонкая схема управления доступом, которая позволяет создавать учетные записи пользователей и затем указывать, например, что для заданных таблиц (или даже заданных столбцов таблицы) пользователь имеет право только на чтение. У нашего пользователя root есть полный доступ ко всему, но вы вряд ли захотите использовать эту учетную запись в реальном web-приложении. Давайте установим *MySQL* и выполним эти действия по-настоящему. *MySQL* есть в репозиториях CentOS, поэтому установка проста:

```
yum install mysql mysql-server
```

Пакет *mysql-server* содержит сам сервер (mysqld), а пакет *mysql* (среди прочего) – клиент командной строки *MySQL*, к которому мы вскоре перейдем.

Также есть обычные команды управления сервисами. Сначала нужно обеспечить запуск сервиса при загрузке системы:



› Набираем меньше, щелкаем больше. *MySQL Workbench* – прекрасная утилита для управления базой данных *MySQL*.

```
# chkconfig mysqld on
```

Затем нужно запустить сервис прямо сейчас:

```
# service mysqld start
```

После этого выполните команду **mysql_secure_installation** для создания начальной конфигурации. Это нужно сделать только один раз. У вас запросят пароль root, но на данном этапе его еще нет, поэтому просто нажмите Enter. Затем вам предложат задать новый пароль root; задайте хороший надежный пароль, потому что у этого пользователя есть полный административный контроль над всей базой данных. Правильнее будет назначить пароль, отличный от пароля root в Linux. Затем, следуя указаниям программы, вы должны:

» Удалить анонимных пользователей.

» Отключить удаленный вход в систему от имени root.

» Удалить тестовую базу данных.

Теперь можно двигаться дальше. Мы пойдем кратчайшим путем, начав прямо с создания базы данных library, которой пользуемся в примерах. Для этого воспользуемся готовым скриптом из команд SQL – *createlibrary-database*. Этот скрипт приведен на рисунке и также должен быть на вашем диске.

В командной строке *MySQL* этот скрипт запускается так:

```
$ mysql -u root -prootpw < create-library-database
```

Здесь **rootpw** – пароль пользователя root *MySQL*, которого мы создали ранее.

Если все пройдет удачно, создастся база данных, и мы сможем выполнять вышеописанные запросы. Зайдите на сервер *MySQL* и выберите базу данных library:

```
$ mysql -u root -prootpw library
```

MySQL напечатает несколько строк сообщений, затем отобразит строку ввода, из которой и выполняются запросы, например:

```
mysql> select author,title from books where onloan = 1;
+----------------+--------------------+
```

»

Снова в школу

Если вы хотите больше узнать об SQL (но не хотите покупать книгу), я бы посоветовал зайти на w3schools.com и щелкнуть по ссылке «Learn SQL» (Изучаем SQL). Здесь вы найдете руководства, список команд SQL и чудесное маленькое приложение, которое позволит набирать SQL-запросы и выполнять их на тестовой базе данных. Если немного расширить наши рамки, на сайте также есть материал для изучения HTML, CSS, PHP, XML и многого другого. Если вы никогда не были на этом сайте, зайдите туда немедля. Обещаю, вы не пожалеете!

```
|  author  |  title  |

+---------------+--------------------+

| Дж. К. Роулинг | Кубок Огня |

+---------------+--------------------+

1 row in set (0.00 sec)
```

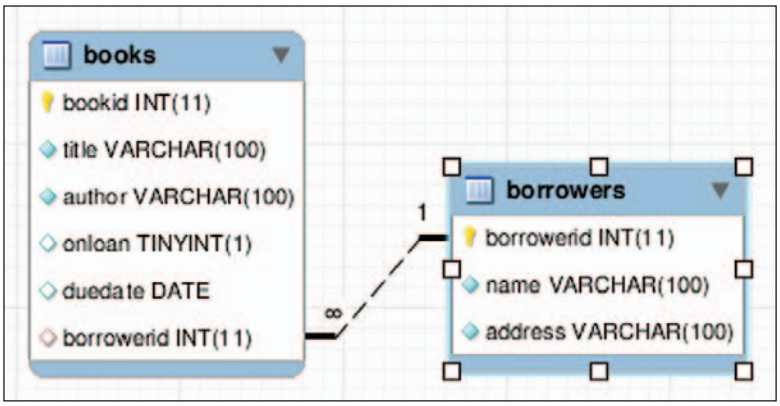
Обратите внимание, что выражения в *MySQL* заканчиваются точкой с запятой. Это часть синтаксиса *MySQL*, а не SQL (если вы похожи на меня, то будете постоянно про эти точки с запятой забывать!).

Другие SQL-запросы и команды, которые мы видели ранее, можно запустить точно так же. Если вы следуете за мной, надеюсь, вам захотелось попробовать выполнить несколько собственных запросов.

Ссылочная целостность

Обычно с *MySQL* используется два серверных движка: MyISAM и InnoDB. В версиях *MySQL* до 5.5 в таблицах по умолчанию использовался MyISAM, а если вам нужна была таблица InnoDB, это указывалось при ее создании. Начиная с *MySQL 5.5* и выше, по умолчанию применяется InnoDB. Одно из преимуществ InnoDB – в принудительном обеспечении ссылочной целостности данных.

«Ссылочная целостность» звучит весьма по-технарски, но по сути это означает, что если внешний ключ существует в таблице, этот ключ (в виде первичного ключа) должен существовать в таблице, на которую ссылается внешний ключ. Например, если бы мы удалили Джейн Смит из таблицы borrowers нашей базы данных library, ссылочная целостность бы нарушилась, потому что ключ этого пользователя (102) указывается в таблице books.



› ***MySQL Workbench*** **строит по базе данных диаграммы «сущность–связь», визуализируя структуру базы.**

Версия Enterprise

Теперь *MySQL* принадлежит Oracle, и программа движется по пути других приложений (вроде *Tripwire*), которые были открытыми, а затем стали коммерческими. Бесплатная открытая версия все еще доступна (и работает отлично), но функциональность уровня предприятия теперь добавляется в платную версию продукта. Эта версия обеспечивает средства «горячего» резервного копирования, репликацию, поддержку отказоустойчивости, масштабируемость, аудит и, разумеется, техподдержку. За подробностями заходите на www.mysql.com/products/enterprise.

Поскольку наша база данных использует движок InnoDB и ссылка внешнего ключа была объявлена в структуре таблицы, при попытке удалить Джейн Смит система вежливо нам откажет:

```
mysql> delete from borrowers where name = 'Jane Smith';
ERROR 1451 (23000): Cannot delete or update a parent row:
a foreign key constraint fails (`library`.`books`,
CONSTRAINT `books_ibfk_1` FOREIGN KEY (`borrowerid`)
REFERENCES `borrowers` (`borrowerid`))
```

Время для кое-чего покрасивее

Если вы устали от командной строки и хотите увидеть что-нибудь графическое, установите *MySQL Workbench*. С этой программой вы можете:

- Создавать схемы баз данных, генерировать базы данных и управлять ими визуально.
- Создавать, оптимизировать и выполнять SQL-запросы.
- Настраивать серверы и учетные записи пользователей.
- Переносить базы данных с SQL Server, Sybase и PostgreSQL.

К сожалению, ее нет в репозиториях CentOS, но ее можно загрузить с сайта *MySQL*. Сделайте следующее:

- Зайдите на www.mysql.com/downloads.
- Щелкните по ссылке *MySQL Workbench*.
- В выпадающем списке выберите RedHat Linux 6.
- Щелкните по ссылке для загрузки 32-битной или 64-битной версии.
- Щелкните по “No thanks, just start my download! [Нет, спасибо, просто начать загрузку!]”, чтобы пропустить регистрацию.
- Сохраните файл (он будет в вашем каталоге **Downloads**).
- Скопируйте файл в **/tmp**.
- Переключитесь на пользователя root и выполните команды:

```
# yum install libzip pexpect gtkmm24
# rpm -i /tmp/mysql*rpm
```

Первая команда установит требуемые зависимости пакета, вторая – сам пакет.

На экранном снимке показан просмотр схемы базы данных в *MySQL* и отображение результатов простого запроса. Учтите, что хотя *MySQL* – графическая программа, вам все равно придется набирать свой SQL-запрос! Мне особенно нравится возможность *MySQL Workbench* по «реинжинирингу» базы данных и построению аккуратной диаграммы «сущность – связь» со схемами таблиц и отношениями между ними.

Через месяц, в последней части серии, мы соединим все части стека LAMP вместе. Мы разработаем web-приложение, которое позволит пользователям библиотеки отправлять запросы к базе данных библиотеки через HTML-форму, и напишем серверный код на PHP, который выполняет эти запросы к базе данных и формирует результат, отформатированный в HTML. Увидимся! LXF

LINUX FORMAT

Учебники

Наши эксперты помогут вам с любым приложением Linux!

ЕВГЕНИЙ БАЛДИН
Подтвердивший свою квалификацию физик.

Kickstarter, игры и Linux

Человек лишь тогда
человек вполне, когда он играет.
Фридрих Шиллер

Чуть менее года назад уже была такая колонка. Тогда я открыл для себя понятие «краудфандинг», или народное финансирование. Это интереснейшее явление вызвано информатизацией общества. Люди платят именно за то, что им нужно. Они рискуют, хотя и малым. Какие-то вложения наверняка разочаруют, но в реальной жизни не все выходит как хочется.

Первое, на что люди тратятся – это зрелища. Зрелища под GNU/Linux в общем есть – само окружение отчасти игрушка для многих, но хотелось бы иметь весь спектр. Поэтому на www.linux.org.ru я целый год раз в две недели делал обзор игровых и не очень проектов для GNU/Linux, имеющих отношение к свободной аппаратуре или свободно распространяемой информации. Лидировали, понятно, игры – и в теле новости, и в обсуждениях. Скоро выяснилось, что для свободных проектов Kickstarter не очень подходит, хотя яркие примеры успешных кампаний были.

А ведя на unixforum.org дайджест «Kickstarter, игры и Linux», я постоянно запрашивал авторов kickstart-проектов: What about Linux support? Год назад около 50% ответов шло в стиле “WTF?”, а сейчас Linux знают все. Нашлись даже его принципиальные враги из философских, а не технических причин. Довольно забавно.

Нас заметили – теперь поборемся!

P.S. *GNU Emacs 23.4* стал лучше.

P.P.S. Мигель де Икаса ушел с Linux на Mac OS X – хорошо иметь выбор...

e.m.baldin@inp.nsk.su

В этом месяце вы научитесь...

Вещать через ТВ 64
Грэм Моррисон берет Raspberri Pi, прикладывает телевизор и *XBMC* и получает Крутое ТВ: смотрите, записывайте, вещайте.

Строить роутер 68
Благодаря роутерам, они же – маршрутизаторы, ваш сетевой трафик попадает куда надо. **Джонатан Робертс** все объяснит.

Спасать систему 72
Нейл Ботвик показывает, как пользоваться отличным аварийным CD для выправления чрезвычайных ситуаций.

Рулить музыкой 74
Шашанк Шарма научит вас организовывать ваши мелодии в фонотеку, не прибегая к графическому интерфейсу.

Засевать облако 78
Присоединяйтесь к **Марко Фиоретти** – он создает собственные онлайн-сервисы на *ownCloud*. Google отдыхает!

Решать проблемы 82
Джоно Бэкон завершает серию рассмотрением ряда профессиональных приемов в *Django*.

Ускорять графику 86
Константин Калгин и **Евгений Балдин** впадают в вендор-зависимость – а что поделаешь? Уж больно технология хороша.

Перевозить офис 90
Максиму Черепанову было лень перетаскивать старые компьютеры с места на место, и он установил виртуальные.

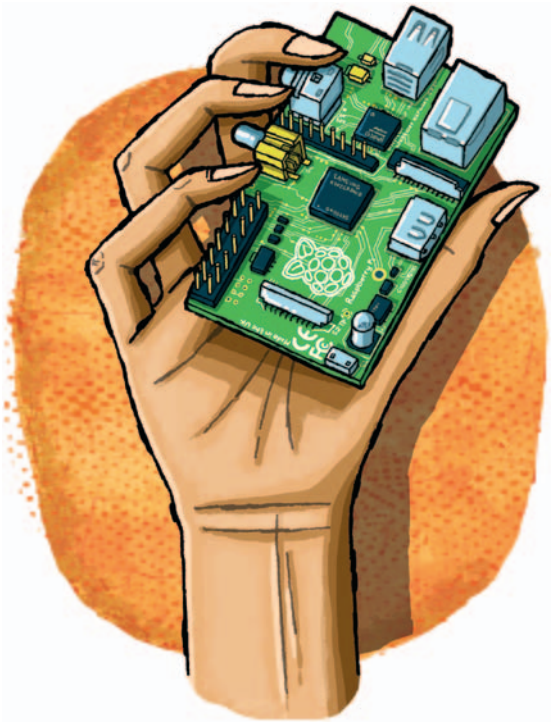
Делать дистрибутив 94
Дмитрий Кузнецов задался целью построить минимальный дистрибутив. Но для этого требуются инструменты. Начинаем...

Raspberry Pi:

После многолетней битвы с *MythTV* **Грэм Моррисон** нашел лучшее решение, благодаря *Tvheadend.*

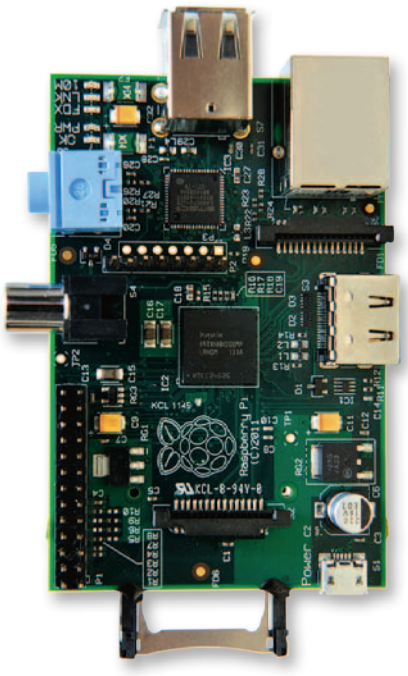
Наш эксперт

Когда **Грэм Моррисон** не ломает голову над тем, что ему поставить в *Linux Format*, и не возится со старыми синтезаторами, он наверняка ищет новые способы повышения продуктивности в KDE 4.



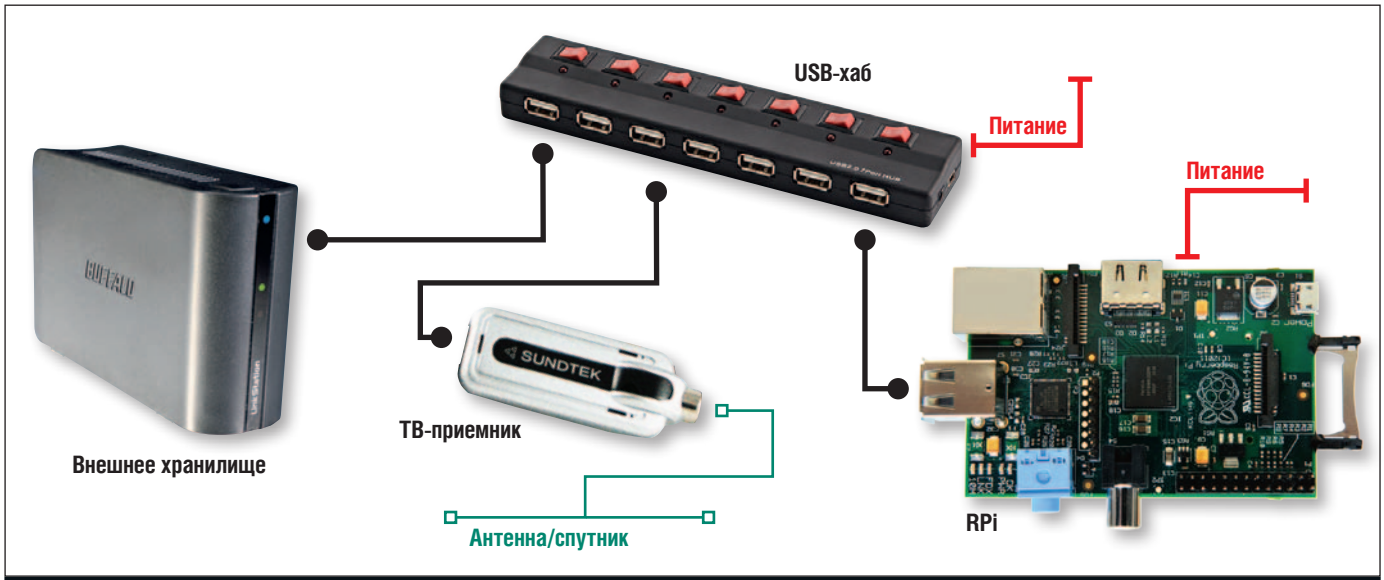
Во многих телевизионных приставках уже есть Linux. И их небольшой вычислительной мощности более чем достаточно для записи и просмотра нескольких каналов одновременно, а также потокового воспроизведения по локальной сети. Raspberry Pi также прекрасно подходит для этих целей, и с подходящим оборудованием его можно превратить в мощный и дешевый цифровой видеомагнитофон, дополненный функциями потокового воспроизведения, расписаний и сдвига времени.

Подходящее оборудование – ключевые слова в предыдущем абзаце, так как безболезненность установки в основном зависит от того, будут ли устройства, предназначенные для записи телевизионного сигнала, «просто работать». К счастью, в Linux поддержка большинства таких устройств уже включена в ядро, поэтому многие устройства будут работать без дополнительных действий. И хотя все начинается в командной строке, мы разбили руководство на 10 шагов, чтобы ему было проще следовать. В конце у вас получится полноценная платформа для записи цифрового ТВ, с возможностью записи нескольких программ из нескольких источников – и все на скромном Raspberry Pi! Это прекрасная «серверная часть» для только что выпущенного медиа-проигрывателя *XBMC*, которым вы сможете воспользоваться в качестве клиента на любом из компьютеров сети.



› ***XBMC*** **(см. обзор на стр. 16) и *Tvheadend* на Raspberry Pi — прекрасное сочетание.**

ТВ-поток



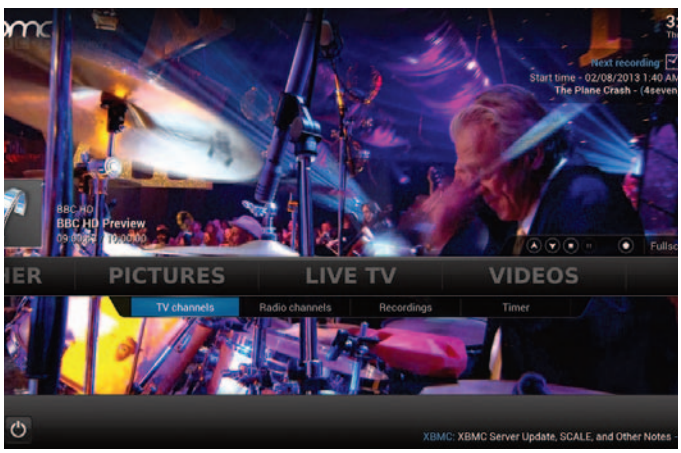
Шаг за шагом

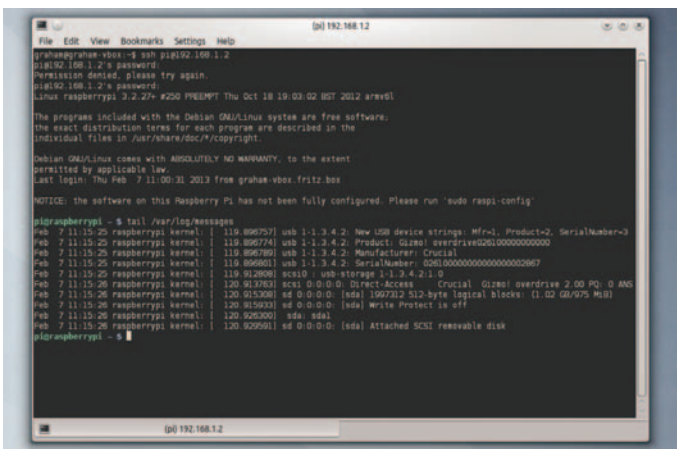
1 Наши устройства

Мы протестировали и настроили два USB-ресивера: один для приема наземного цифрового телевидения через антенну, а второй – для получения данных со спутника. Инструкции приводятся и для того, и для другого. Для приема сигнала DVB-T (наземного) мы воспользовались Sundtek MediaTV Pro, для DVB-S (спутникового) – Sundtek SkyTV Ultimate. Для последнего нужен блок питания 12 В, который также нужно подключить. Но вот самое важное требование: эти USB-устройства должны быть подключены к Raspberry Pi через USB-хаб с внешним питанием. Мы потратили два дня в попытках настроить систему – сначала без хаба, потом с несовместимым хабом. В обоих случаях казалось, что все работает, но при сканировании не находилось ни одного канала. Хаб с внешним питанием, совместимый с Raspberry Pi, решил проблему, так что это очень важно. Подключите к хабу источник питания, затем USB-ресивер, а потом подключите хаб к Pi. И не забудьте подключить кабель от антенны или спутниковой тарелки к ресиверу.

2 Внешнее хранилище

Мы предполагаем, что на Raspberry Pi есть настроенный и обновленный дистрибутив Raspbian. Мы также предполагаем, что Pi подключен к Интернету, и что вы набираете команды напрямую в консоли или через SSH. Теперь о том, где хранить записи. Рекомендуем подключить внешний USB-диск, так как постоянное чтение и запись на SD-карту чревато ее быстрым выходом из строя. Чтобы добавить внешний жесткий диск, просто подключите его к свободному порту USB и проверьте сообщения в системном лог-файле, набрав **tail /var/log/messages**. Вы увидите сообщение вроде **usb 1-1.3.4.2: New USB device**, и нужно обратить внимание на идентификатор устройства, который должен выглядеть примерно так: **sda: sda1**, где **sda** – само устройство, а **sda1** – раздел. Наберите **sudo mkdir/mnt/storage**, чтобы создать точку монтирования, и **sudo mount /dev/sda1 /mnt/storage/**, чтобы связать ее с внешним жестким диском.





» **Не хотите пропустить номер?** Подпишитесь на www.linuxformat.ru/subscribe/!

3 Установка драйверов

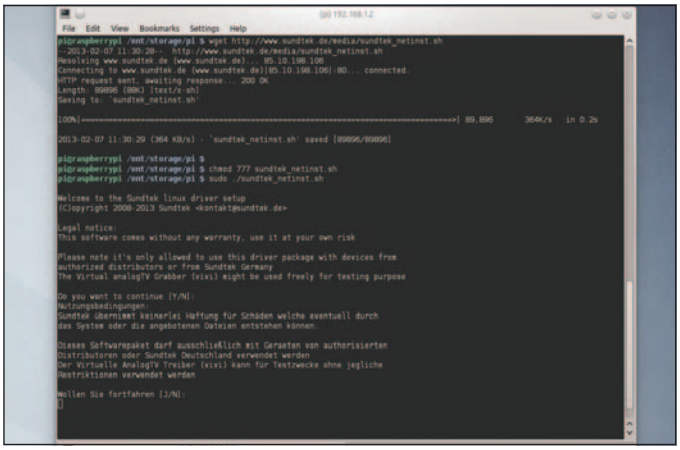
В зависимости от используемого оборудования, этот шаг может быть необязательным. Если вы выбрали устройство, которое совместимо с Linux и не требует дополнительных файлов драйверов, можно просто подключить устройство и перейти к следующему шагу. Для наших устройств Sundtek драйвер нужно загрузить и установить. Это легко. В командной строке Raspberry Pi наберите

```
wget http://www.sundtek.de/media/sundtek_netinst.sh
chmod 777 sundtek_netinst.sh
sudo ./sundtek_netinst.sh
```

Последняя строка запустит скрипт, который мы загружаем в первой строке. Он определит версию системы и загрузит последнюю версию драйверов. Драйверы останутся запущенными и настроенными на запуск при загрузке системы. Пользователям DVB-T также нужно выполнить следующую команду:

```
/opt/bin/mediaclient --setdtvmode=DVBT
```

Она гарантирует, что устройство настроено на прием наземного сигнала, а не сигнала от кабельного источника, поскольку устройство умеет делать и такое.



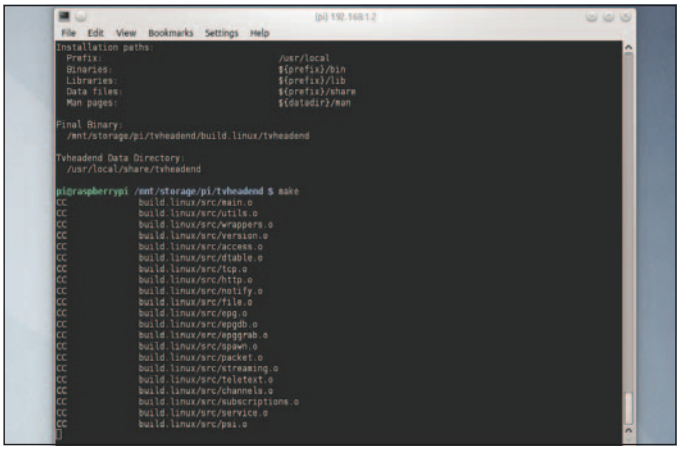
4 Установка Tvheadend

Программа, которой мы будем пользоваться для записи и потокового воспроизведения телевизионного сигнала, называется *Tvheadend*. С помощью модуля расширения для недавно вышедшего *XBMC* можно превратить этот превосходный медиа-проигрыватель в полноценный цифровой видеомагнитофон, а *Tvheadend* будет делать всю трудную работу в фоновом режиме на Raspberry Pi. Так как *Tvheadend* активно разрабатывается, мы воспользовались нестабильной версией, но если вы хотите пропустить этот шаг, можно так же легко воспользоваться дистрибутивом OpenElec вместо Raspbian. К счастью, сборка программы проста. Сначала установите утилиты для разработки и DVB:

```
sudo apt-get update
sudo apt-get install unzip libcurl4-openssl-dev pkg-config git build-essential dvb-apps gcc-4.7
```

После этого загрузите последнюю версию *Tvheadend* из репозитория разработчиков через *git*, и для ее сборки воспользуйтесь троицей **./configure**, **make** и **sudo make install**:

```
CC=gcc-4.7; ./configure; make; sudo make install
```

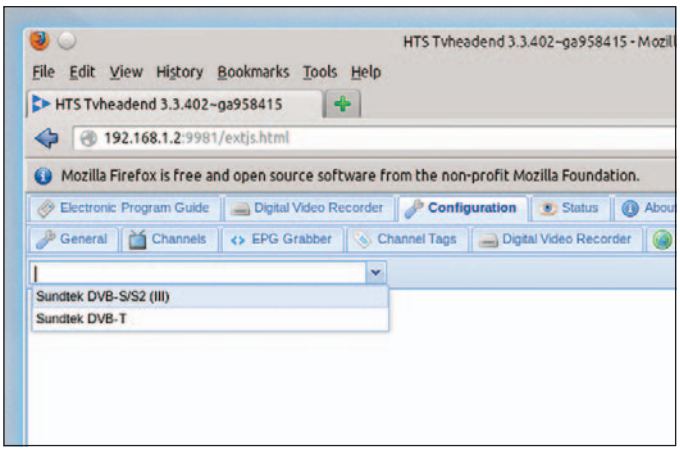


5 Настройка Tvheadend

Пользовательский интерфейс *Tvheadend* доступен через web-браузер, но сначала его нужно запустить. Так как мы осуществляем запуск в первый раз, сделаем это в режиме настройки, и запустим его как демон, то есть он станет фоновым процессом. Чтобы это сделать, наберите

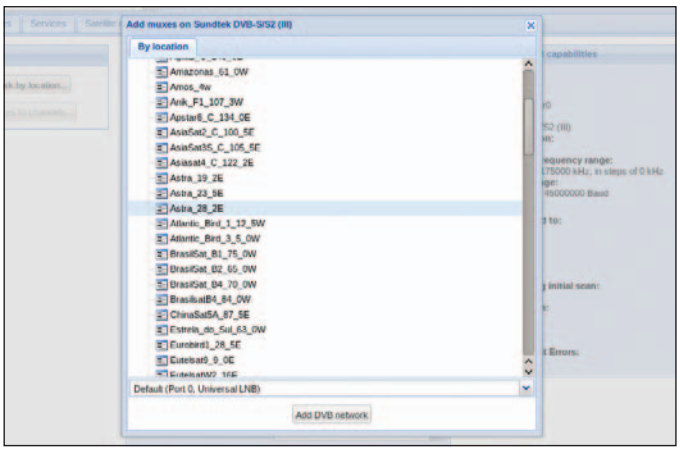
```
tvheadend -C -d
```

Теперь откройте браузер, желательно на другом компьютере сети, и введите http://ip_of_rpi:9981/extjs.html. IP-адрес Raspberry Pi можно узнать, набрав **ifconfig** и посмотрев на значение поля 'inet addr' устройства 'eth0'. Браузер загрузит клиент по умолчанию для *Tvheadend*. В нем вы, наконец, увидите данные всех программ, а также настройки и просмотра записей. Нам нужно сказать клиенту, как пользоваться телевизионным оборудованием, которое мы подключили. Для этого выберите Configuration > TV Adaptors [Настройка > ТВ-Адаптеры] и выберите свое устройство из выпадающего списка слева. Следующие шаги будут зависеть от того, наземный или спутниковый сигнал вы принимаете, поэтому мы их разделим. Для спутникового сигнала переходите к следующему шагу, для наземного – к шагу 7.



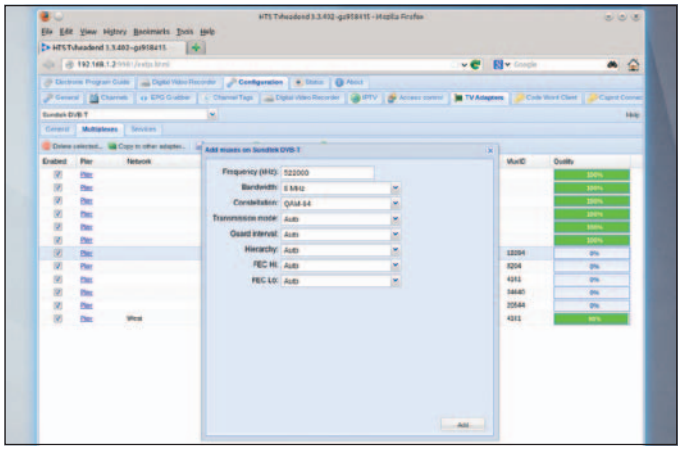
6 Настройка приема спутникового сигнала

Прием спутникового сигнала настроить проще всего. Если адаптер выбран, то на странице General [Общее] появится обзор конфигурации. Поставьте галочку Enable [Включить] и нажмите Save [Сохранить]. Мы также добавим информацию о каналах, для чего вводится информация по поиску спутника и мультиплексам, которые он предоставляет, затем на этих мультиплексах производится поиск каналов, доступных для приема. *Tvheadend* соберет данные о спутнике по его местоположению, поэтому достаточно нажать кнопку Add DVB Network by location [Добавить сеть DVB по местоположению] слева. Тогда откроется окно с глобальным списком спутников. Для Великобритании и Северной Европы лучший вариант – Astra_28_2E. После выбора спутника *Tvheadend* добавит список мультиплексов для этого спутника на вкладку Multiplexes [Мультиплексы]. Они будут просканированы на имеющиеся каналы, и ход сканирования будет доступен в разделе Capabilities [Ресурсы] на странице General [Общее] справа. Если все работает, вы увидите обнаруженные сервисы (каналы) на вкладке Services [Сервисы]. Для Astra_28_2E мы обнаружили 970 сервисов из 98 мультиплексов. Если вы не хотите настроить систему также и для приема наземного сигнала, перейдите к шагу 8.



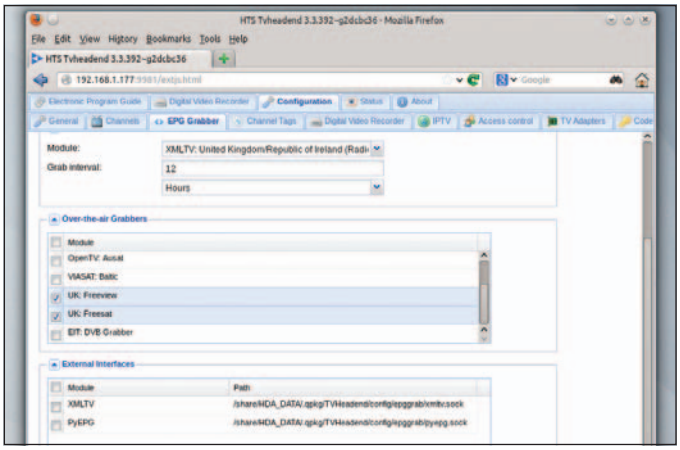
7 Настройка приема наземного сигнала

Сначала активируйте приемник на главной странице. *Tvheadend* содержит обширную базу данных настроенных передатчиков с подробной информацией о каждом мультиплексе. Вам нужно будет узнать, на какой передатчик указывает ваша антенна. В Великобритании это удобнее всего сделать на сайте www.digitaluk.co.uk/postcodechecker. Когда узнаете, просто нажмите кнопку Add DVB Network [Добавить сеть DVB] на вкладке General [Общее] адаптера и найдите передатчик «По расположению [By location]». Нашего местного передатчика в списке не оказалось. Пришлось зайти на www.ukfree.tv/transmitters.php, ввести индекс и добавить информацию о каждом мультиплексе вручную. Это можно сделать на вкладке Multiplexes [Мультиплексы] адаптера с помощью кнопки Add mux(es) manually [Добавить мультиплекс(ы) вручную]. В появившемся окне нужно ввести частоту [frequency], частотную полосу [bandwidth] и группу [constellation] для каждого мультиплекса, а все остальное установить в Auto [Автоматически]. Эти данные можно получить на сайте. Обычно это нужно делать для трех или четырех мультиплексов, в зависимости от вашего местоположения. *Tvheadend* просканирует их на сервисы/каналы и добавит в вашу конфигурацию.



8 Добавление каналов

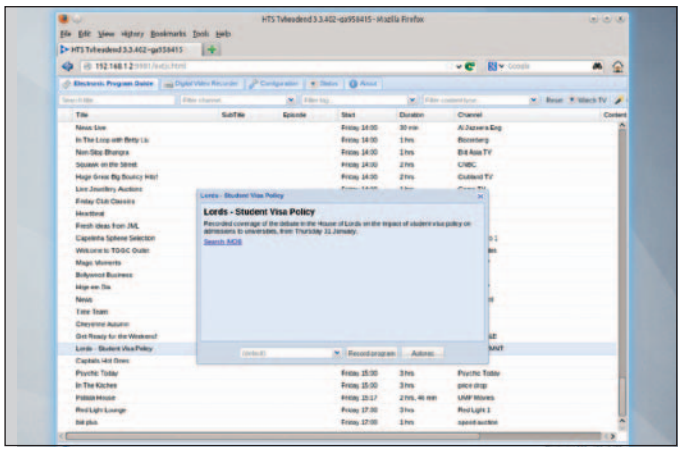
Теперь у нас есть большой список сервисов, полученный из мультиплексов. Следующий шаг – разрешить *Tvheadend* получить из этих сервисов каналы; делается это на вкладке General [Общее] адаптера нажатием кнопки Map DVB Services To Channels [Связать сервисы DVB с каналами]. После этого у вас должен получиться список каналов для просмотра на странице Channels [Каналы] вне окна настройки адаптера. Каталог размещения записей по умолчанию нужно изменить – вероятно, на точку монтирования, которую мы создали в начале. Для этого зайдите на страницу Digital Video Recorder [Цифровой видеомагнитофон] и измените путь Recording System [Система записи]. Чтобы изменения вступили в силу, нужно сохранить настройки на этой странице. Также нужно настроить электронный телегид. Обычно у каждого канала есть минимум в виде What's On Now [Сейчас в эфире] и What's On Next [Далее в эфире], но сервисы вроде Freeview и Freesat в Великобритании предоставят более полную телепрограмму на семь дней. Чтобы активировать эти сервисы, зайдите на страницу EPG Grabber [Сервис электронного телегида], затем щелкните на любом из сервисов в разделе Over-The-Air-Grabbers [Эфирные сервисы]. После этого не забудьте сохранить конфигурацию, нажав Save Configuration.



9 Запись программы

Вскоре на странице Electronic Program Guide [Электронный телегид] появятся эфиры – их можно посмотреть или добавить в расписание для записи. Щелчок на любой программе откроет окно настройки записи. Другой вариант, Autorec [Автоматическая запись], интереснее: выполняется поиск по одним и тем же данным программ, чтобы записать все эпизоды, не полагаясь на ссылки из телегида.

В зависимости от числа каналов и объема данных телегида страница может сильно разрастись. Чтобы решить эту проблему, можно отфильтровать отображаемые данные, используя строку с опциями в верху списка. Можно искать по фрагменту заголовка или ограничиться одним каналом или тэгом фильтра. Если вы нашли фильтр, который вас устроил, по нажатию кнопки Create Autorec [Создать автоматическую запись] параметры поиска будут добавлены в *Tvheadend*, которая затем будет записывать все, что соответствует результатам поиска. Удалить запланированные записи можно на вкладке Digital Video Recorder [Цифровой видеомагнитофон]. Предстоящие записи удаляются с первой страницы, а фильтры Autorec – с последней. На срединных страницах можно воспроизвести или удалить сделанные записи или проверить, почему программа не записалась.



10 Просмотр записей и эфира

Установив плагин *VLC* для браузера, можно просматривать записи и эфирное ТВ в браузере. Мы проверили это только в *Firefox*, но по щелчку на эфирной программе появится возможность воспроизвести его [Play]. Если плагин *VLC* не установлен, вам предложат установить его; после установки должно появиться окно с эфиром программы. Кнопки в верху этого окна могут перевести в полноэкранный режим или сделать паузу в текущем эфире. Точно так же можно просматривать записанные программы на странице Digital Video Recorder [Цифровой видеомагнитофон]. Если вы не хотите пользоваться браузером, скопируйте и вставьте сетевой адрес в *VLC* на другом компьютере или узнайте в одном из руководств, как организовать общий доступ к папкам с записями по сети. Но на наш взгляд, *Tvheadend* удобнее всего пользоваться с новой версией *XBMC*, где есть плагин, способный общаться с *Tvheadend* напрямую, загружая телегид с Raspberry Pi и позволяя просматривать эфирные каналы, а также планировать и просматривать записи. Настроить его легко – просто зайдите на страницу плагина PVR для *XBMC*; мы планируем через месяц подробно рассказать об *XBMC*, и даем вам это как задание на дом! LXF

ЧАСТЬ 2

Сеть: Построим

Познакомив вас с кирпичиками, из которых строят сети, **Джонатан Робертс** складывает из них реально работающую сеть.

Наш эксперт

Джонатан Робертс сбежал из Башен *Linux Format*, чтобы поискать счастья в качестве сисадмина.

Месяц назад мы познакомили вас с основными понятиями сетей – пакетами, физическими соединениями и логической адресацией, а также с утилитами, при помощи которых в Linux можно исследовать работу сети. Пользуясь ими, вы сможете эффективно работать с сетями и диагностировать проблемы, с которыми столкнетесь.

На данном уроке мы покажем, как эти элементы сочетаются друг с другом в настоящих сетях. Сперва заставим два компьютера взаимодействовать друг с другом напрямую, затем продемонстрируем, как превратить компьютер с Linux в маршрутизатор (он же – роутер). Парк компьютеров вам не потребуется – все это мы будем делать в *VirtualBox*, но вы должны уметь применять те же самые технологии к настоящим сетям.

› **Содержимое каталога** /etc/sysconfig/network-scripts **выглядит устрашающе, но, сосредоточившись на файлах** ifcfg-ethX**, вы увидите, что с ними можно сделать многое.**

```
root@Adam:/etc/sysconfig/network-scripts
[root@Adam network-scripts]# ls
ifcfg-Auto_Ethernet              ifdown-post     ifup-plusb
ifcfg-Auto_Ethernet-1            ifdown-ppp      ifup-post
ifcfg-Auto_greateranglia-wifi    ifdown-routes   ifup-ppp
ifcfg-Auto_HTC_Portable_Hotspot  ifdown-sit      ifup-routes
ifcfg-Auto_LANEWLESS             ifdown-tunnel   ifup-sit
ifcfg-Auto_netgear               ifup            ifup-tunnel
ifcfg-eth0                       ifup-aliases    ifup-wireless
ifcfg-lo                         ifup-bnep       init.ipv6-global
ifdown                           ifup-eth        keys-Auto_HTC_Portable_Hotspot
ifdown-bnep                      ifup-ippp       keys-Auto_LANEWLESS
ifdown-eth                       ifup-ipv6       keys-Auto_netgear
ifdown-ippp                      ifup-ipx        net.hotplug
ifdown-ipv6                      ifup-isdn       network-functions
ifdown-isdn                      ifup-plip       network-functions-ipv6
[root@Adam network-scripts]#
```

В наших примерах мы воспользуемся CentOS – он основан на Red Hat Enterprise Linux и является популярной платформой для серверов. Если вам интересен Ubuntu или другие дистрибутивы на основе Debian, загляните во врезку «Дистрибутивы на базе Debian».

Прежде чем перейти к сети, создадим систему, в которой мы будем практиковаться. Сначала установите *VirtualBox*. Он должен быть в репозиториях большинства дистрибутивов, так что для его установки просто воспользуйтесь менеджером пакетов своего дистрибутива. После этого загрузите минимальный установочный диск CentOS 6.3 (http://bit.ly/1446qQO).

Теперь создайте две новых виртуальных машины (ВМ) в *VirtualBox*, следуя нашему пошаговому руководству на соседней странице, позаботившись добавить все необходимые сетевые интерфейсы.

Затем повторите те же шаги для второй ВМ, убедившись, что она подключена к той же внутренней сети. Создав виртуальные машины с двумя такими сетевыми интерфейсами, как у нас, мы сделаем так, что они обе смогут подключаться к Интернету (через NAT по умолчанию), но также смогут и взаимодействовать друг с другом напрямую по внутренней сети, которую мы добавили. Эта внутренняя сеть эквивалентна подключению двух виртуальных машин к одному коммутатору [switch] или простому соединению их сетевым кабелем.

Базовая настройка сети

Завершив установку на обоих компьютерах, подключим обоих к Интернету через сеть NAT по умолчанию. Это даст нам возможность познакомиться с сетевыми конфигурационными файлами Red Hat и освежить кое-что, о чем мы говорили в прошлый раз.

После загрузки виртуальной машины первым делом надо перейти в каталог **/etc/sysconfig/network-scripts**. В дистрибутивах на базе Red Hat этот каталог содержит скрипты запуска и файлы настройки всех сетевых устройств, подключенных к компьютеру. Если вы следовали нашему пошаговому руководству, вы должны увидеть два конфигурационных файла – **ifcfg-eth0** и **ifcfg-eth1** (в других версиях Red Hat это могут быть **ifcfg-em1** и **ifcfg-em2**). **Eth0** – первое сетевое устройство, и оно должно ссылаться на первый адаптер (Adapter 1) в настройках *VirtualBox* – это сеть NAT.

Откройте в *Vi* файл **ifcfg-eth0** и убедитесь, что в нем есть такие параметры со своими значениями:

```
NM_CONTROLLED="no"
BOOTPROTO="dhcp"
ONBOOT="yes"
```

Первая строка говорит о том, что менеджер сети (NetworkManager) не должен управлять этим интерфейсом – раз мы пытаемся делать это вручную, его вмешательство может внести путаницу, поэтому мы его отключаем.

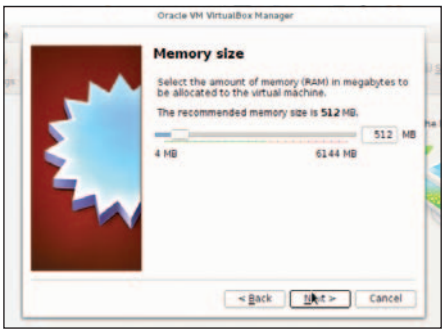
Вторая строка определяет, какой протокол должен использоваться для получения IP-адреса. Здесь мы указали DHCP (Dynamic Host Configuration Protocol – протокол динамической

маршрутизатор

Создаем виртуальную машину

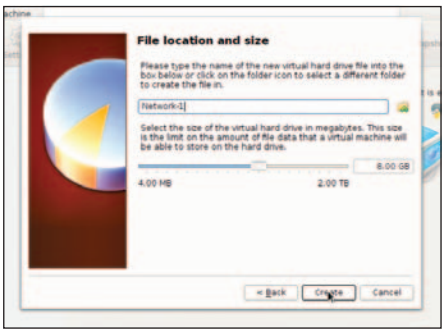
1 Дадим своей ВМ имя

Нажмите на кнопку Создать [New] в левом верхнем углу окна VirtualBox. Задайте имя ВМ и выберите Linux и Red Hat (вместе с соответствующей архитектурой) из выпадающих списков.



2 Выберем объем памяти

Задайте объем памяти в 512 МБ – для простейших сетевых задач это более чем достаточно.



3 Создадим виртуальный диск

Размер может быть любым, но не менее 8 ГБ. Динамически назначенный диск сэкономит время при создании виртуальной машины, но может работать медленнее.

4 Добавим сеть

Выделив новую ВМ, выберите Settings [Настройки], затем Network [Сеть]. Переключитесь на Adapter 2 [Адаптер 2], поставьте галочку Enable [Включить] и убедитесь, что он подключен ко внутренней сети.

5 Выберем диск

В разделе Storage [Устройство хранения данных] меню Settings [Настройки] выберите Empty [Пустой] рядом с IDE controller [Контроллер IDE], затем щелкните на иконке CD справа и откройте загруженный ISO-образ CentOS.

6 Установим CentOS

Наконец, нажмите Start [Начать] и пройдите процедуру установки CentOS.

настройки хоста). Это означает, что ВМ получит IP-адрес, шлюз и информацию о DNS-серверах автоматически с удаленного сервера. В данном случае это не удаленный сервер, а *VirtualBox*, который эмулирует его.

Возможны и другие значения параметра **BOOTPROTO**, в том числе static, если вы хотите вручную назначить этому компьютеру постоянный IP-адрес.

Наконец, **ONBOOT** говорит, что интерфейс должен запускаться при каждом запуске скриптов инициализации сети на этом компьютере, то есть при загрузке системы или перезапуске сетевого сервиса. Проверьте это сами, выполнив от имени root команду **/etc/init.d/network restart**, и убедившись, что сеть работает так, как должна. Для этого можно попробовать отправить пакеты на популярный сайт, например, на Google, командой **ping**: **ping google.com**.

» **Не хотите пропустить номер?** Подпишитесь на www.linuxformat.ru/subscribe/!

NAT

В этой статье мы упомянули нечто под названием NAT и соответствующую концепцию, называемую маскированием. Наверное, вам будет полезно узнать, что же такое NAT, особенно если учесть, что вы почти наверняка уже пользуетесь им дома.

NAT – сокращение от “Network Address Translation [Преобразование сетевого адреса]”. Чтобы понять, что это означает на самом деле, представьте себе свое домашнее интернет-подключение. Ваш провайдер выдает вам внешний IP-адрес, посредством которого сайты и другие компьютеры в Интернете могут общаться с вами. Но почти наверняка больше одного адреса провайдер вам не даст.

» **Проблема нехватки адресов IPv4** Причина в том, что сегодня адресов IPv4 осталось мало. Всего их 4294967296, и с виду это огромное количество, но первоначально адреса выделялись блоками и всем желающим, что привело к быстрому сокращению их запасов. Именно поэтому ваш провайдер ограничился выдачей вам одного IP-адреса.

Но, как и у большинства людей, у вас дома скорее всего несколько устройств, подключенных к Интернету: у вас вполне может быть несколько мобильных телефонов, несколько ноутбуков, планшет, настольный компьютер и телевизор Smart TV. А у технарей их и того больше! Как они все подключаются к Интернету, если IP-адрес всего один?

» **Тайные адреса** Ответ – NAT. Этот единственный IP-адрес назначается роутеру, который в свою очередь назначает уникальные, частные IP-адреса всем устройствам в доме. Когда одно из этих устройств хочет подключиться к Интернету, подключение производится через шлюз по умолчанию – ваш роутер. Затем роутер изменяет содержимое пакетов, заменяя частный IP-адрес публичным адресом роутера. После этого он отправляет пакет в Интернет к месту назначения. Заодно он следит за тем, какой пакет относится к какому устройству, и при получении ответа снова изменяет адрес получателя. Но на этот раз он перенаправляет ответ извне на исходное устройство, начавшее диалог.

На нашем уроке первая виртуальная машина выступает как роутер для второй, выполняя преобразование сетевого адреса для ее пакетов и отправляя их во внешний Интернет. Хитрость здесь и в том, что первая виртуальная машина тоже находится за собственным NAT *VirtualBox*, который, вероятно, тоже находится за NAT роутера.

Немного везения, и ваши пакеты вернутся – это значит, что вы подключены к Интернету, следовательно, ваши IP-адрес, маска подсети (или префикс), шлюз по умолчанию и DNS – все, что нужно для подключения подсети – заданы верно и работают правильно.

Просмотреть все эти настройки можно, заглянув в различные конфигурационные файлы или с помощью утилит, о которых мы говорили в прошлый раз. Например, команда **ip route** выведет шлюз по умолчанию (в данном случае, виртуальный шлюз, предоставляемый *VirtualBox*); в **/etc/resolv.conf** можно найти сервер имен; а команда **ip a** выведет ваши IP-адрес и префикс. Повторите все эти действия для второго компьютера.

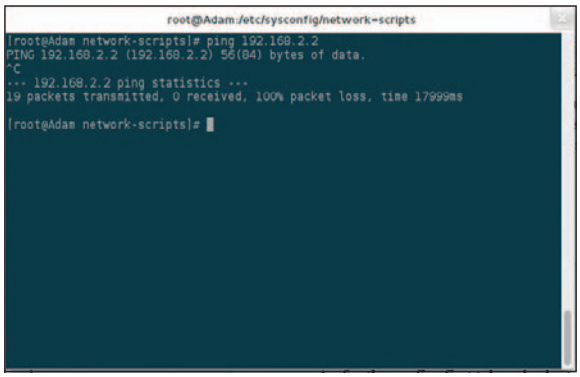
Статические адреса

В предыдущем разделе большую часть трудной работы сделали протокол DHCP и *VirtualBox*. Теперь мы настроим сетевое соединение между двумя виртуальными машинами, но все настройки будем добавлять сами.

Первое, что нужно отметить – если бы это были настоящие компьютеры, их бы пришлось соединять физическим кабелем, либо напрямую, либо через свитч или роутер. Так как это виртуальные машины, мы уже сделали это, подключив их вторые интерфейсы к одной и той же внутренней сети.

Поэтому, как и прежде, начнем с каталога **/etc/sysconfig/network-scripts**. Вам понадобится файл **ifcfg-eth1**, но он будет выглядеть не так, как в прошлый раз:

```
NM_CONTROLLED=”no”
```



› **Попытавшись попинговать хост в другой по отношению к вам сети, вы получите в ответ лишь молчание.**

```
BOOTPROTO=”static”
IPADDR=”192.168.1.2”
PREFIX=24
ONBOOT=”yes”
```

Протокол загрузки теперь установлен в static [статический] – это означает, что вместо автоматического получения IP-адреса нам придется задать его вручную. Это делается в параметрах **IPADDR** и **PREFIX**. IP-адрес может быть любым, но принято, чтобы он начинался с **10** (8-битный префикс), **172.16** (12- битный префикс) или **192.168** (16- битный префикс). Эти адреса выделены для «частного» использования, и у большинства роутеров они закрыты для доступа извне. Кроме того, не забывайте, что первый (для 24-битного префикса – **.0**) и последний (**.255**) адреса зарезервированы как адрес сети и широковещательный адрес, поэтому их назначать нельзя.

Чтобы выбрать префикс, взгляните на врезку «Подробнее про IP-адреса» из прошлой статьи и затем примите решение. Помните, что для 24-битного префикса только последний октет идентифицирует хост, а остальные – сеть; для 16-битного префикса, два последних октета представляют хост, а первые два – сеть. Мы воспользуемся 24-битным префиксом.

После этого сбросьте сеть, как раньше.

Затем попытайтесь попинговать со второй виртуальной машины первую по ее новому IP-адресу: **192.168.1.2**. Ни один из пакетов не должен вернуться. Чтобы соединение можно было установить, нужно дать второму компьютеру IP-адрес с точно таким же префиксом (или маской подсети). Мы советуем использовать **192.168.1.3**. Добавьте его в соответствующий файл **/etc/sysconfig/network-scripts** и перезапустите сеть.

Если теперь попинговать адрес **192.168.1.2** со второго компьютера, пакеты должны возвращаться. Более того, если перейти на первый компьютер и попробовать попинговать **192.168.1.3**, это тоже должно работать. Подключение по SSH также должно работать, и если запустить web-сервер на одной из виртуальных машин, он должен быть доступен с другой.

Теперь два компьютера могут взаимодействовать. Отлично! Но при каждом подключении набирать IP-адреса неудобно. Обычно компьютерам назначают имена, которые проще запомнить и легче набирать. Это можно сделать несколькими способами. Например, в сети с большим количеством компьютеров стоит создать свой сервер DNS (Domain Name System –

» **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

система доменных имен), но это выходит далеко за рамки нашего урока.

Так как у нас всего два компьютера, их имена можно задать гораздо проще – поместить их в файл **/etc/hosts**. Этот файл содержит список IP-адресов и их алиасов, поэтому в него можно добавить записи для двух наших компьютеров:

```
192.168.1.2 lxf-network-1
192.168.1.3 lxf-network-2
```

Если добавить их в файл **/etc/hosts** на обоих компьютерах, вы должны увидеть, что компьютеры пингуются по именам **lxf-network-1** и **lxf-network-2**, и вас перенаправляют на нужный компьютер – так-то лучше.

Создаем роутер

Пока мы не задавали шлюз ни на одном из наших компьютеров, поскольку получали его от DHCP из сети NAT *VirtualBox* по умолчанию вместе с данными DNS, и таким образом наши компьютеры подключались к внешнему Интернету. Чтобы продемонстрировать еще кое-какие возможности сети, отключим соединение NAT на второй виртуальной машине и настроим первую как роутер для предоставления доступа в Интернет.

Сначала выполним команду **ifdown eth0** для выключения интерфейса NAT на второй виртуальной машине. Теперь, когда интерфейс настроен, эта команда вместе со своей напарницей **ifup** может использоваться для включения и выключения интерфейсов.

Потом попробуйте попинговать Google с этого компьютера, и вы увидите, что это больше не работает. Чтобы все заработало снова, нужно внести изменения в конфигурацию первой виртуальной машины.

Сначала ей нужно разрешить перенаправлять пакеты с одного сетевого интерфейса на другой – это самое важное и единственное, для чего нужен роутер. В Linux такое можно, но по умолчанию эта функция отключена. Чтобы снова ее включить, откройте файл **/etc/sysctl.conf** – он управляет различными аспектами работы ядра, и установите **net.ipv4.ip_forward = 1**. Затем запустите **sysctl -p /etc/sysctl.conf** для загрузки новой конфигурации.

После этого вам понадобится настроить собственный NAT, или сетевое маскирование; в Linux это делается с помощью IPTables – брандмауэра, встроенного в ядро.

IPTables пугают многих, но на самом деле все очень просто: это лишь набор правил, на которые смотрит ядро при обработке пакета. Если пакет соответствует любому из этих правил, ядро применяет к нему действие, указанное в правиле. Действия могут включать отбрасывание или прием пакета или манипуляцию с ним для изменения его источника или места назначения.

В нашем случае, когда пакет прибывает на первую виртуальную машину со второй на интерфейс **eth1**, мы хотим применить

```
root@Adam:/etc/sysconfig/network-scripts
[root@Adam network-scripts]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTA
BLISHED
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     udp  --  anywhere             224.0.0.251         state NEW udp dpt:
mdns
REJECT     all  --  anywhere             anywhere            reject-with icmp-h
ost-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  anywhere             anywhere            reject-with icmp-h
ost-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@Adam network-scripts]#
```

к нему действие **masquerade** – то есть направить его во внешний мир через интерфейс **eth0**, представив, что пакет пришел с первой, а не со второй виртуальной машины:

```
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

Разберем это:

- **-t nat** определяет, на какой набор правил в какой таблице правил нужно посмотреть.
- **-A POSTROUTING** – поднабор правил, которые нужно просмотреть, иначе известный как цепочка. Цепочка определяет, в какой точке процесса маршрутизации должно применяться правило.
- **-o eth1** определяет, что это правило должно применяться ко всем пакетам, которые будут выходить из интерфейса **eth1**.
- **-j MASQUERADE** говорит, что нужно «перепрыгнуть» на действие **MASQUERADE**, когда любой пакет соответствует этому правилу.

С этим правилом добавьте шлюз по умолчанию **GATEWAY = 192.168.1.2** в файл **/etc/sysconfig/network** (в котором хранится общая сетевая конфигурация для всех интерфейсов) на второй виртуальной машине, задайте популярный внешний сервер имен в **/etc/resolv.conf**, добавив туда **nameserver 8.8.8.8** (IP-адрес сервера имен Google), и у вас должно получится полностью рабочее интернет-подключение, несмотря на то, что никакого прямого подключения у вас нет.

Если вы хотите подробнее узнать об IPTables, прочтите несколько HowTo на сайте Netfilter: http://netfilter.org/documentation/.

Вот и все, на что у нас хватило места в этом месяце. Но не отчаивайтесь: за последние два урока вы многое узнали о сети. Проявив немного терпения, вы сможете читать другие руководства и создавать гораздо более сложные конфигурации, точно соответствующие вашим потребностям. LXF

› IPTables могут выглядеть сложно, особенно когда правил много, но в простых случаях добавить несколько правил очень легко.

Дистрибутивы на базе Debian

В этой статье мы занимались конфигурацией, специфичной для Red Hat, а в CentOS используется именно эта модель. Конечно, дистрибутивы на базе Red Hat – не единственное достойное решение для серверов; многие предпочитают Debian или его более молодой и популярный вариант Ubuntu.

Однако в дистрибутивах на базе Debian используется другой подход. Все утилиты, о которых мы вам рассказали – набор команд **ip**, **ping** и т.д. – здесь работают. Более того, файлы вроде **/etc/resolv.conf** и **/etc/hosts** находятся в тех же местах.

Зато различие есть в файлах настройки интерфейса. Вместо разбиения конфигурации на отдельные файлы, такие как **ifcfg-eth0** и **ifcfg-eth1** в Red Hat, в дистрибутивах на базе Debian используется файл **/etc/network/interfaces**, где находятся настройки всех интерфейсов. Вот пример этого файла:

```
auto eth0
iface eth0 inet static
address 192.168.1.2
netmask 255.255.255.0
```

Первая строка **auto eth0** говорит, что интерфейс **eth0** нужно включить при выполнении команды **ifup -a** или при загрузке системы.

Следующие параметры, начиная с **iface**, содержат подробные параметры конфигурации интерфейса. Сначала указывается интерфейс, затем протокол подключения. Здесь возможны несколько вариантов, но самый популярный – **inet**, означающий IP-сети.

Слово **static** – это «метод» получения адреса; другие варианты – **dhcp** и **bootp**. Методы могут принимать дополнительные параметры, и они приводятся далее с отступами под заголовком. Наш пример очень простой, в нем задаются IP-адрес и маска подсети для этого интерфейса.

Более подробную информацию и возможные параметры можно найти на странице **man /etc/network/interfaces**.

System Rescue:

Нейл Ботвик представляет небольшой полезный Live-дистрибутив, который придет к вам на помощь при серьезном кризисе в системе.

Наш эксперт

У **Нейла Ботвика** по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не скажет вам, где находится центральный сервер.

Обширное сообщество разработчиков Linux достойно благодарности за многие новшества. Одно из самых популярных, но, возможно, недооцененных – дистрибутив Live CD. С тех пор, как Клаус Кноппер [Klaus Knopper] выпустил свой первый Live CD Knoppix 13 лет назад, для запуска Linux без установки системы и даже без рабочего жесткого диска было достаточно вставить компакт-диск в подходящий привод и перезагрузиться. Теперь в большинстве дистрибутивов Live CD или DVD используется как основа для установки, но они также очень полезны, если что-то пошло наперекосяк и система перестала загружаться. Для этого можно обойтись и установочными дисками, но они громоздки и содержат массу лишнего, поэтому для них часто нужен DVD или объемистая флэшка. По-настоящему удобным был бы небольшой дистрибутив, который можно записать на компакт-диск или USB-брелок и всегда иметь под рукой на случай аварии.

Такой компакт-диск для спасения системы есть, и это SystemRescueCD! Это мой любимейший из многих Live-дистрибутивов (в противоположность дистрибутивам для установки систем на настольный компьютер). Последний ISO-образ для такого небольшого пакета весит 350 МБ. Он содержит массу полезных программ для восстановления систем Linux и Windows. И все это поместится на любой брелок, да еще останется достаточно места на более традиционное применение брелка – например, для переноса файлов между компьютерами.

Опции загрузки

Загрузите последнюю версию образа с www.sysresccd.org. На время написания статьи это 3.1.2, но System Rescue CD часто обновляется, и когда вы будете это читать, вероятно, будет уже более свежая версия. Запишите ее на компакт-диск как обычно, и загрузитесь с него, чтобы увидеть первоначальное меню загрузки. Опций загрузки многовато, но по умолчанию они вполне разумны. SystemRescueCD поставляется с 32-битным и 64-битным ядрами. 64-битную систему можно загрузить с обоих, так как все программы на диске 32-битные, но если вам нужно запускать из установленной системы 64-битные программы, например, изменить корневой каталог с chroot, чтобы решить какую-то проблему, воспользуйтесь 64-битным ядром. В последних версиях нужное ядро определяется автоматически по типу процессора, поэтому в большинстве случаев можно выбрать готовый вариант. Однако есть несколько дополнительных команд, которые можно добавить в процесс загрузки; чтобы увидеть все опции, нажимайте клавиши с F2 по F7.

Для каждого ядра есть два варианта загрузки – если стандартное ядро не загружается, попробуйте альтернативное. Среди интересных опций – docache, которая загружает весь образ в оперативную память, то есть после загрузки компьютера вы сможете извлечь диск и вставить другой для восстановления системы. С опцией setkmap вам не будут предлагать выбрать раскладку клавиатуры, а rootpass позволяет задать пароль root. Они особенно удобны в собственной версии дистрибутива (подробнее об этом позже), так как с ними вы сможете загрузить сервер без монитора с компакт-диска и подключиться к нему по SSH.

› С помощью клавиш от F2 до F7 можно просмотреть подробную информацию о других опциях загрузки, и еще больше опций загрузки есть в документации.

```
F1 | F2 | F3 | F4 | F5 | F6 | F7        SystemRescueCd          [boot options]

* You can add the following boot options after the image name:
You must add these options after the boot image (F2 for images). You can use
the default boot image name (rescue32) in most cases. For instance if you
want to load the whole system into the memory type this: rescue32 docache

docache     copy system in RAM: you can eject the cd, requires 512 MB of RAM
setkmap=x   configure the keyboard with the x keymap (x='fr', 'de', 'us', ...)
root=x      root=/dev/sda1 boots from /dev/sda1, root=auto scans the disks
forcevesa   use the Xvesa grahical server instead of Xorg when you type startx
noload=x    prevent a module from being loaded at startup (eg: noload=3c59x)
doload=x    force to load one/several modules at startup (eg: doload=3c59x)
rootpass=x sets the root password to x so that you can ssh to the system
ethx=ip     automatically configure the ethernet interfaces with an ip address
gateway=ip automatically set the default route in the network configuration
dns=ip      automatically configure the ip address of the DNS server to use
dostartx    Runs the graphical environment automatically at boot time
minishell   Run the debug shell from the initramfs during the boot process

* Press F2 for the list of the boot images (use rescue32 by default)
* Press F5 for boot options related to boot problems and troubleshooting.
* Press F7 for the list of the other options related to the network
```

Лучшая оболочка

После загрузки системы вы попадете в командную строку (SystemRescueCD использует превосходную оболочку Zsh, но пользователи *Bash* почувствуют себя в ней как дома): ведь для диска восстановления системы нам не нужен графический интерфейс. Ах, нужен? Тогда наберите в командной строке **startx**, либо при загрузке выберите седьмой пункт меню. В нем к списку опций загрузки добавляется dostartx, и все так и сделается само.

Пользуетесь ли вы графическим интерфейсом или консолью, помните, что при загрузке системы вы заходите в нее полноправно, от имени root. Поэтому справедливы обычные предупреждения – вы сможете разрушить систему, и если сделаете это, любуйтесь на обломки. Это также значит, что каждый, у кого есть SystemRescueCD, может получить доступ администратора. Если компьютер находится в небезопасном месте, как большинство ноутбуков, отключите загрузку с USB или CD в BIOS и задайте пароль для BIOS, чтобы никто не мог эти настройки вернуть.

Компакт-диски несколько громоздки – флэшки гораздо удобнее, особенно если у вас тонкий ноутбук без оптического привода. Чтобы скопировать SystemRescueCD на USB-брелок, загрузитесь с компакт-диска, вставьте брелок подходящего объема и выполните команды

```
/livemnt/boot
./install.sh
```

Можно это сделать и без загрузки с компакт-диска (что удобно, если у вас нет оптического привода), смонтировав ISO-образ из обычного дистрибутива.

```
sudo mount -o loop systemrescuecd-x86-3.1.2.iso /mnt/tmp
cd /mnt/tmp
sudo ./usb_inst.sh
cd
sudo umount /mnt/tmp
```

Live-ремонт

Этот скрипт пишет данные в файловую систему USB-брелка. В отличие от утилит-аналогов, он оставляет все не нужное ему место доступным для обычного использования. Одно из преимуществ копирования на брелок в том, что здесь легко менять опции загрузки по умолчанию. Например, чтобы убрать паузу, возникающую, когда в процессе загрузки у вас запрашивается раскладка клавиатуры, откройте **syslinux/syslinux.cfg** на брелке и добавьте **setkmap=uk** (ну, или какая вам нужна раскладка) к опциям APPEND всех пунктов меню, которым вы будете пользоваться. Также можно изменить настройку MENU DEFAULT, чтобы по умолчанию выделялся другой пункт меню. Аналогично, изменив параметры TIMEOUT и ONTIMEOUT, можно изменить время отображения меню перед загрузкой пункта меню по умолчанию, и какой пункт меню будет выбираться. Параметр TIMEOUT задается в десятых долях секунды.

Из закона Мэрфи следует, что дисков восстановления системы никогда нет под рукой, когда они нужнее всего, а USB-брелок легко потерять. С SystemRescueCD можно быть всегда готовым к этой ситуации, так как его можно загрузить прямо из ISO-образа на жестком диске, если у вас *Grub 2*. Скопируйте файл ISO в **/boot** (**/boot** мы взяли, потому что это единственная часть файловой системы, гарантированно доступная при загрузке с *Grub*), затем создайте файл в **/etc/grub.d**, например, с именем **40_sysrescd**, содержащий следующее:

```
menuentry “System Rescue CD - Std kernel” {
set sysresiso=/sysrescd.iso
loopback loop $sysresiso
linux (loop)/isolinux/rescue64 setkmap=uk
isoloop=$sysresiso
initrd (loop)/isolinux/initram.igz
}
```

Имя файла – это имя файла ISO-образа; я копирую файл в **/boot** и затем создаю символическую ссылку на **sysrescd.iso**, чтобы не менять меню *Grub*. Выберите в качестве ядра rescue32 или rescue64 – автоматическое определение [autodetection] здесь недоступно, да и не нужно. Затем сделайте файл исполняемым и скомандуйте

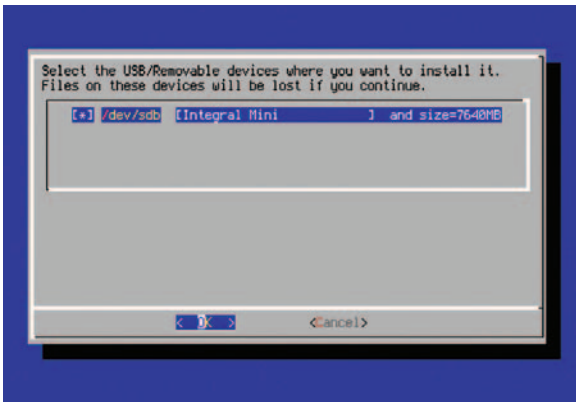
```
sudo update-grub2
```

Теперь у вас в меню загрузки есть SystemRescueCD. Подумайте о том, о чем мы говорили выше – бесконтрольном root-доступе к компьютеру, если он физически не защищен. Возможно, стоит добавить пароль в *Grub*, чтобы никто, кроме вас, не мог загрузить этот пункт меню.

Восстановление системы с chroot

Ранее мы упомянули о chroot; это способ изменения корневого каталога, так что можно работать внутри операционной системы, даже не загрузившись в нее. Например, предположим, что ваш друг забыл пароль root для своей системы. Это означает, что, ко всеобщему сожалению, вы не можете войти в систему и сбросить его. Можно загрузиться с Live CD, но напрямую изменить пароль не получится, поэтому загрузитесь с SystemRescueCD и выполните следующие команды (здесь **/dev/sdxN** содержит корневой раздел вашего дистрибутива):

```
mount /dev/sdxN /mnt/custom
mount -t proc none /mnt/custom/proc
```



› Установщик USB создаст загрузочный брелок из компакт-диска или его ISO-образа.

```
mount --bind /dev /mnt/custom/dev
cp /etc/resolv.conf /mnt/custom/etc/
chroot /mnt/custom /bin/bash
```

Мы смонтировали ваш корневой каталог в **/mnt/custom**, сделали файловые системы **/dev/** и **/proc** доступными в этом каталоге (программа с chroot-доступом не может обращаться к файлам вне chroot) и скопировали **/etc/resolv.conf** в chroot на случай, если вам понадобится интернет-доступ. Последняя команда заходит в chroot, запуская указанную команду для появления оболочки. Теперь вы по сути зашли в свой дистрибутив, не загрузив его, и можно выполнять большинство команд, таких как

```
passwd
```

для сброса пароля root. Закончив, наберите **logout** (или нажмите Ctrl+D) для выхода из chroot-оболочки, размонтируйте то, что смонтировали ранее – и перезагрузитесь.

Добавление собственных возможностей

Сайт SystemRescueCd, www.sysresccd.org, содержит массу информации по добавлению собственных возможностей, и на самом диске есть несколько скриптов, которые помогут в этом. Однако в большинстве случаев это, скорее всего, не понадобится. Программ, имеющихся на диске, достаточно для подавляющего большинства ситуаций, даже если у вас зашифрованные файловые системы в томах LVM в RAID-массиве. Вам может понадобиться добавить немного данных, а это проще сделать с USB-устройством. USB-брелки ныне достаточно дешевы, чтобы на одном можно было уместить систему восстановления и резервные копии. Все, что вам нужно – установить на брелок SystemRescueCD, это очистит брелок. Затем создайте на брелке каталог, скажем, **data**, чтобы файлы были отдельно, и скопируйте туда необходимые файлы. После загрузки в Live-окружение ваши файлы будут доступны в каталоге **/livemnt/boot/data**, если вы не пользовались опцией docache; в этом случае придется смонтировать брелок вручную самим.

Если при каждой загрузке с SystemRescueCD приходится выполнять одни и те же команды, поместите их в скрипт *autorun* в корне USB-брелка. Можно задать несколько таких файлов для различных ситуаций, привязав их к различным пунктам меню загрузки. Более подробно об этом и обо многих других аспектах использования и настройки SystemRescueCD см. на сайте. LXF

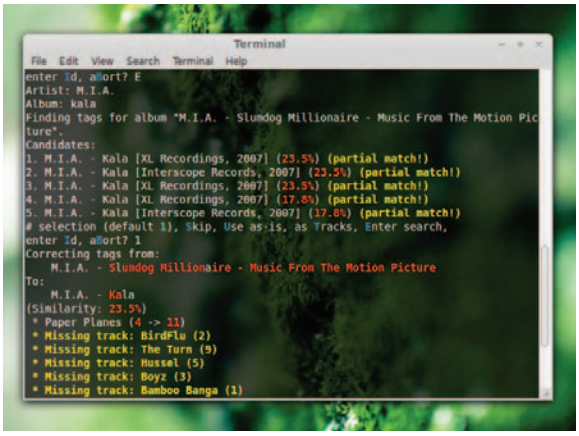
Beets: Рулите

Технари-меломаны ликуют. Фанат командной строки **Шашанк Шарма** открыл менеджер файлов мультимедиа для командной строки.

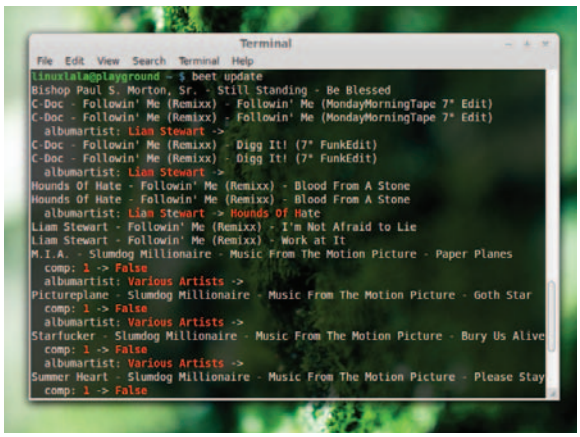
Наш эксперт

Шашанк Шарма более четырех лет пишет о свободном ПО для различных изданий включая Linux.com. Он соавтор книги «Fedora для начинающих».

Недавние тектонические сдвиги в популярных рабочих столах привели к тому, что многие люди меняют пристрастия, рабочие среды и операционные системы. А командная строка не спеша пыхтит себе в полной уверенности, что будет только привлекать новых пользователей, но никогда не потеряет старых. Причина – отчасти в огромном количестве мощных утилит, которые она предоставляет.



› Если у вас большая фонотека, скопируйте несколько файлов в отдельный каталог и импортируйте их в *Beets*, чтобы увидеть автоматический тэггер в действии. Это грандиозное зрелище!



› После добавления или удаления файлов в каталоге с музыкой можно обновить библиотеку командой beet update**.**

Здесь мы обсудим *Beets*, менеджер медиа-библиотеки, умеющий исправлять метаданные песен, загружать обложки, тексты, чинить имена файлов и др. В отличие от других аналогов из командной строки, *Beets* немного неуклюж – по крайней мере, в названии. Все потому, что он отвергает традиционное для командной строки совпадение имени приложения с именем команды. Чтобы воспользоваться *Beets*, вы запускаете не **beets**, а **beet**! Установить *Beets* можно из менеджера пакетов дистрибутива, но версия скорее всего будет старой. Проект активно разрабатывается, и лучше загрузить самую последнюю версию.

Независимо от вашего дистрибутива, последнюю версию *Beets* удобнее всего установить с *pip* – это система управления пакетами, используемая для установки пакетов Python и управления ими. Это не альтернатива *yum* и другим подобным утилитам. Если *pip* еще не установлен в вашей системе, установите пакет *python-pip* из программных репозиториев. Команда

```
sudo apt-get install python-pip
```

или

```
su -c 'yum install python-pip'
```

сделает это, а также установит зависимости, такие как *python-setuptools* и *python-dev*. Затем установите *Beets* командой

```
pip install beets
```

Pip загрузит и установит все необходимые зависимости. В дальнейшем можно обновить *Beets* в любой момент, командой

```
pip install -U beets
```

Параметр **-U** велит *pip* обновить установленный пакет. Если вы устанавливали пакет из менеджера пакетов дистрибутива, не пытайтесь обновить его с *pip*. Если вы это сделаете, то у вас будет два отдельных экземпляра одного и того же пакета. То же относится к удалению пакета. Если вы устанавливали пакет с *pip*, удаляйте его командой

```
pip uninstall packagename
```

Самая обширная задача при работе с *Beets* – создание его файла настройки, которого по умолчанию не существует. Его

своей музыкой

придется создать вручную. Но пусть это вас не беспокоит. Понадобится минута или две, во всяком случае, для начала. Откройте свой любимый текстовый редактор и введите

```
[beets]
directory: ~/Music
library: ~/beetsmusic.blb
```

Сохраните файл под именем **.beetsconfig** в своем домашнем каталоге. Это минимально необходимое содержимое файла **~/.beetsconfig**. Файл библиотеки уже есть в домашнем каталоге пользователя, но при желании его можно передислоцировать в другое местоположение.

В зависимости от настройки Beets, параметр directory выше служит одной из двух целей. Это либо каталог, в котором хранятся ваши музыкальные файлы, либо пустой каталог, куда вы собираетесь переместить свои файлы после того, как их проиндексирует *Beets*.

Если вы намерены только импортировать файлы в библиотеку Beets, сохранив текущие структуру каталогов и расположение файлов, укажите в параметре directory каталог со своими музыкальными файлами и выполните команду

```
beet import ~/Music
```

Она проиндексирует все музыкальные файлы из указанного каталога. В порядке альтернативы предположим, что ваши музыкальные файлы хранятся в каталоге **~/Desktop/songs/**, а вы хотите импортировать и переместить их в каталог **~/Music**. Тогда прежде всего сделайте так, чтобы параметр directory в файле **~/.beetsconfig** указывал на пустой каталог. Также нужно добавить в конец файла четвертую строку:

```
import_move: yes
```

Она гарантирует, что при запуске команды

```
beet import ~/Desktop/songs
```

музыкальные файлы из каталога songs будут проиндексированы и перемещены в каталог **~/Music**. Если не указывать

```
import_move: yes
```

Beets скопирует все проиндексированные файлы в пустой каталог **music**. Если вы не хотите и копировать файлы, добавьте в файл **~/.beetsconfig** строку

```
import_copy: no
```

Параметры настройки

Полный список параметров слишком обширен, чтобы приводить его полностью, но мы обсудим несколько важных параметров по ходу статьи. В документации проекта есть информация о многих параметрах, и мы настоятельно советуем сначала прочесть ее.

Пусть обилие параметров вас не пугает: начиная пользоваться *Beets*, всех их указывать не нужно. Настройки по умолчанию удовлетворяют большинство пользователей, поэтому поместить три вышеупомянутые строки в **~/.beetsconfig** будет достаточно для старта в *Beets*.

Мы коснулись импорта файлов лишь наскоро, но это самый важный аспект менеджера медиа-библиотеки, поэтому займемся им поподробнее.

Загружаем тексты песен

С помощью плагина lyrics можно загрузить тексты для ваших музыкальных файлов, и его можно настроить так, чтобы тексты добавлялись прямо к метаданным самого файла.

Сначала подключите плагин, приписав его в **~/.beetsconfig** к строке

```
plugins =
```

После этого *Beets* автоматически загрузит тексты для всех треков во время импорта.

Если в **~/.beetsconfig** задан параметр

```
import_write: yes
```

тексты еще и запишутся в базу данных, в противном случае *Beets* сохранит тексты в собственной базе.

В случае, если вам захочется отключить автоматическую загрузку текстов во время импорта, добавьте в файл **~/.beetsconfig** раздел из двух строчек, следующего содержания:

```
[lyrics]
autofetch: no
```

Почти у всех плагинов есть собственные параметры, и у каждого из них должен быть свой уникальный заголовок в файле **~/.beetsconfig**.

Для ручной загрузки текстов песен, которые уже имеются в базе данных *Beets*, вы также можете написать собственный запрос **query-foo**.

При импортировании файлов в библиотеку *Beets* попытается найти информацию о файлах в Интернете и получить всевозможные метаданные этих файлов.

Затем он попытается добавить метаданные в файлы. В зависимости от количества добавляемых в библиотеку файлов, это потребует некоторого времени. Команда

```
beet import /path/to/music/files
```

выполняет несколько функций. Она добавит файлы в библиотеку, скопирует файлы в каталог, указанный в **~/.beetsconfig**, и попытается загрузить и добавить в файлы метаданные и другие тэги. Если у вас большая коллекция музыки, и вы не хотите добавлять в них тэги при импорте в библиотеку, укажите параметр **-A**, чтобы *Beets* не добавлял тэги автоматически:

```
beet import -A /path/to/music/files
```

Возможность автодобавления тэгов может прославить или опозорить *Beets* в будущих релизах. Пока ею лучше не пользоваться, если у вас на диске сотни треков. В нашем тесте импортирование восьми песен с включенным автоматическим добавлением тэгов

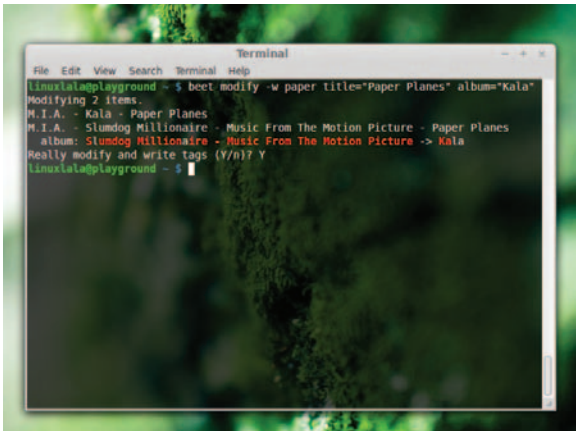
Скорая помощь

Если в ключевых словах в запросах есть пробелы, заключите их в кавычки или экранируйте:

```
beet ls pink\ floyd
```

или

```
beet ls “pink floyd”
```



Изменить метаданные любого файла в библиотеке можно командой

```
beet modify
```

Для вывода списка всех опций изменения метаданных выполните команду

```
beet ? modify
```

» **Не хотите пропустить номер?** Подпишитесь на www.linuxformat.ru/subscribe/!

заняло несколько минут. Более того, эта возможность работает только с альбомами. То есть, если вся ваша музыка лежит в одном большом каталоге, *Beets*, возможно, не сделает многого для каждого из этих файлов. Однако если вы приложили некоторые усилия и разместили треки в разных каталогах, *Beets* сочтет каждый каталог альбомом и попробует найти метаданные для треков. Если вы хотите только загрузить метаданные, не применяя их к файлам, нужно воспользоваться ключом **-W**. Команда

```
beet import -W /path/to/files
```

велит *Beets* загрузить метаданные и сохранить их в базе данных без записи в музыкальные файлы. Вместо использования параметра **-W** можно добавить параметр

```
import_write: yes
```

или

```
import_write: no
```

в файл **~/.beetsconfig**. Еще одна важная возможность при импорте файлов – возобновление, **resume**. Она активируется, если *Beets* обнаружит прерванный **import**. Импорт может быть прерван пользователем или в случае ошибки. *Beets* фиксирует все прерванные импорты и при обнаружении незавершенного импорта спрашивает у вас, что нужно сделать. Если вы хотите, чтобы все прерванные импорты всегда восстанавливались в исходное состояние, можно добавить строку

```
import_resume: yes
```

в файл **~/.beetsconfig** или, если вам лень, воспользоваться параметром **-p**. Команда

```
beet import -p /path/to/files
```

восстановит все прерванные импорты, если это возможно. Строка

```
import_resume
```

в **~/.beetsconfig** принимает три возможных значения: yes [да], no [нет] и ask [спросить]. Есть множество других похожих параметров, применяемых при импорте файлов в *Beets*. Полный список и краткое описание всех параметров можно получить командой

```
beet ? Import
```

Скорая помощь

Для получения краткой статистики проиндексированных файлов, такой как количество треков, общее время воспроизведения и т.д., пригодится команда

```
beet stats
```

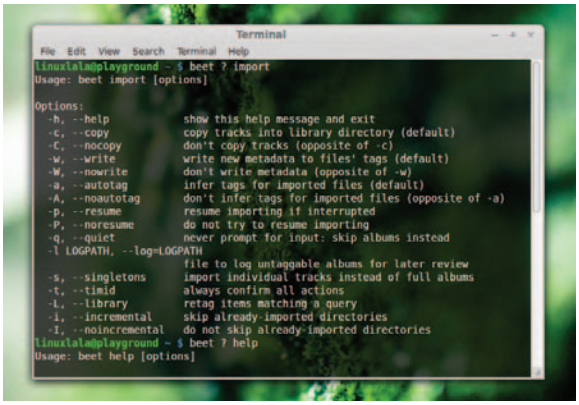
Скорая помощь

Назначение параметров в верхнем и нижнем регистре в *Beets* различно и является противоположным: например, **-w** (записывать) и **-W** (не записывать), **-p** (восстанавливать) и **-P** (не восстанавливать).

Работа с автоматическим тэггером

Если файлов у вас немного или файлы тщательно отсортированы по различным каталогам, вы должны попробовать автоматический тэггер *Beets* в действии.

Сделать так, чтобы у всех файлов были осмысленные имена, нельзя, и вполне естественно, если в имя файла закрадется спецсимвол или у песни нет точного названия. Если это касается и ваших музыкальных файлов, автотэггер не только загрузит метаданные для файлов, но и попытается переименовать файлы, заодно очистив их имена от лишних символов.



› Если параметры в ~/.beetsconfig противоречат параметрам, указанным при запуске *Beets*, настройки в файле игнорируются.

Для получения всей информации о файлах *Beets* подключается к базе данных *MusicBrainz*. Затем для каждого файла в альбоме *Beets* попросит подтвердить, что найденная информация верна. Он выведет список всех треков, найденных в Интернете, с вероятностью соответствия каждого в процентах.

Если все файлы распределены по каталогам, у автотэггера Beets не возникнет с ними проблем, даже если они глубоко упрятаны в каталоги вроде **songs/hiphop/male/2000s/fanboy-artist-hero/**. *Beets* также распознает альбомы на нескольких дисках и помечает их вместе как один релиз. Для этого он ищет ключевые слова, такие как ‘part’, ‘vol.’, ‘volume’, ‘CD’ и ‘disc’.

Для каждого найденного варианта *Beets* отображает процент соответствия. Соответствие в 100 % – точное, и в этом случае *Beets* при необходимости переименует файл автоматически. Если процент соответствия не слишком высок или соответствие имени файла не полное, а частичное, *Beets* попросит вас подтвердить изменения.

```
linuxlala@playground ~ $ beet import Music/Infinite_Livez/Cracks\ in\ the\ Great\ Lens/
/home/linuxlala/Music/Infinite_Livez/Cracks in the Great Lens
Finding tags for album “Infinite Livez - Cracks in the Great Lens”.
Candidates:
1. INFINITE - INFINITIZE [Woollim Entertainment, 2012] (36.5%)
2. INFINITE - OVER THE TOP [Woollim Entertainment, 2011] (35.1%) (partial match!)
3. Infinite Livez vs. Stade - Morgan Freeman’s Psychedelic [Big Dada, 2008] (33.2%) (partial match!)
4. Annie Lennox - Pavement Cracks (The Remixes) [BMG UK & Ireland, 2003] (29.8%)
5. Infinite Livez - Bush Meat [2004] (27.9%) (partial match!)
# selection (default 1), Skip, Use as-is, as Tracks, Enter search, enter Id, aBort?
```

В данном случае *Beets* не смог найти точного исполнителя/альбом и предоставил список частичных соответствий. Можно либо пропустить импорт этого альбома, нажав S, либо нажать U, чтобы импортировать треки как они есть, с текущими метаданными. Если файлы не являются частью альбома, нажмите T. Затем *Beets* попытается найти для них соответствие среди синглов, а не альбомов. Также можно сообщить *Beets* исполнителя и название альбома, по которым он сможет найти информацию в базе данных в Интернете. Если ни один из этих вариантов вас не удовлетворяет, нажмите B, чтобы не импортировать этот альбом в вашу упорядоченную библиотеку.

Иногда Beets не слишком уверен в найденных вариантах и предлагает поискать другие:

Графический Beets

В *Beets* также есть плагин web-gui, хотя он и находится на экспериментальной стадии разработки. Чтобы его активировать, добавьте **web** к строке

```
plugins =
```

в файле **~/.beetsconfig**.

Теперь выполните команду

```
beet web
```

и откройте в браузере http://localhost:8337, чтобы увидеть графический интерфейс *Beets*. В текстовом поле в левом верхнем углу можно вводить поисковые строки. Для воспроизведения треков из библиотеки плагин использует HTML5 Audio. Просто дважды щелкните на песне, и *Beets* воспроизведет ее для вас.

При желании имя хоста и порт можно задать командой

```
beet web [HOSTNAME] [PORT]
```

или прописать эту информацию в файле настройки **~/.beetsconfig**:

```
[web]
host=127.0.0.1
port=5555
```

Учтите, что плагин еще в разработке и может неожиданно «упасть».

» **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

```
linuxlala@playground ~ $ beet import Music/The\ Strangers/
/home/linuxlala/Music/The Strangers
```

Здесь тэги

```
The Strangers - Live On WFMU's Minor Music with Jesse
Krakow, 11-19-12
```

были изменены на

```
Strangers - Shine On You
(Similarity: 47.1%)
* A Little Too Fast (3) -> Shine On You (1)
* Missing track: Shine On You (Draper remix) (2)
[A]pply, More candidates, Skip, Use as-is, as Tracks, Enter search,
enter Id, aBort?
```

При нажатии M *Beets* выведет список других возможных совпадений. Затем можно выбрать трек из списка, если найден верный вариант, или импортировать треки как есть. Если совпадения не найдено, можно нажать A для применения изменений, и *Beets* переименует файлы.

Запросы к библиотеке

Невозможно, да и незачем помнить все треки/альбомы, которые вы индексировали в библиотеке *Beets*: к библиотеке в любой момент можно легко сделать запрос командой **list**. Например, команда

```
beet ls desire
```

выведет все песни со словом 'desire' в имени файла. *Beets* будет искать слово 'desire' в метаданных всех файлов и отобразит результаты, в которых это слово есть в любых метаданных, в названии, исполнителе, названии альбома и т. д.

Если указать несколько слов, *Beets* автоматически объединит их логическим «и» и выведет только те результаты, в которых есть все указанные слова в любом порядке. То есть, одно из слов может быть в имени исполнителя, другое – в названии альбома, и т. д. Все слова могут быть в разных полях. Если результатов слишком много, уточните запрос, задав поле поиска. Например, команда

```
beet ls artist:pink
```

покажет все результаты, в которых имя исполнителя соответствует 'pink'. Точно так же можно задавать параметры альбомов параметром **-a** с командой

```
beet ls
```

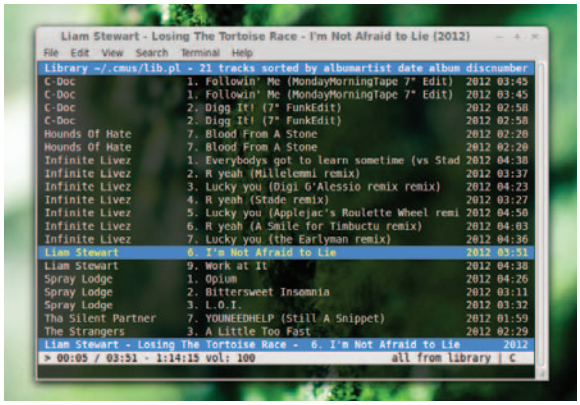
Список всех альбомов заданного жанра выведет команда

```
beet ls -a genre:rap
```

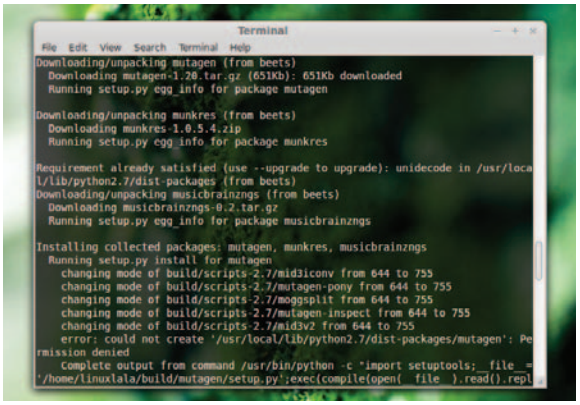
В одном запросе можно объединять несколько полей, чтобы сузить рамки поиска.

В поисковых запросах можно использовать и несколько полей, например, year [год], genre [жанр], month [месяц] и country [страна]. Команда

```
beet fields
```



› В музыкальных проигрывателях для командной строки нет недостатка, но *CMUS* с его элегантным дизайном — лучший.



› Лучше не пользоваться менеджером пакетов дистрибутива и *pip* одновременно. Можно что-нибудь испортить.

выведет полный список полей трека и альбома, которые можно использовать в запросах. Аналогичная структура запроса есть у команды **remove**, так что можно писать сложные запросы для удаления записей из библиотеки *Beets*. Например, если вы хотите удалить из библиотеки файлы со словом 'infin', скомандуйте

```
beet remove infin
```

Затем *Beets* попросит вас подтвердить, что вы хотите удалить все показанные файлы, которые соответствуют запросу. Чтобы удалить файлы и из библиотеки, и с диска, добавьте к команде **remove** параметр **-d**. Команда

```
beet remove -ad infi
```

удалит из библиотеки и с диска все альбомы (параметр **-a**), в которых есть слово 'infi'.

Модули расширения Beets

Основное назначение *Beets* – помочь вам упорядочить музыкальные файлы. Он не пытается делать что-то еще. Тем не менее, как у любой приличной программы, у *Beets* существует добавочный список модулей-плагинов, расширяющих его функциональность.

Нет ничего плохого, если программа для управления музыкальным каталогом заодно проигрывала бы музыку, даже не имея всех возможностей полноценного музыкального плейера, и разработчики *Beets*, похоже, с этим согласны. *Beets* поставляется вместе с *BPD*, а это, как можно догадаться по названию, клон популярного музыкального проигрывателя командной строки *MPD*. В нем нет всей функциональности *MPD*, и вам не обязательно проигрывать музыку в *BPD* или упорядочивать музыкальные файлы с *Beets*. Сама документация проекта указывает на *CMUS*, один из самых простых и мощных музыкальных проигрывателей в командной строке, которые только можно найти.

На самом деле, нам не удалось заставить *BPD* работать с *Beets* ни на одном из тестовых компьютеров с разными дистрибутивами! Сам процесс настройки *BPD* состоит из нескольких этапов, поэтому лучше пользоваться *CMUS* или любым клиентом *MPD*.

По умолчанию в *Beets* есть несколько модулей расширения, но сначала их нужно активировать. Для активации модуля зайдите в файл **~/.beetsconfig** и добавьте строку

```
plugins =
```

В этой строке задается разделенный пробелами список модулей, которые нужно активировать. Скажем, если вы хотите активировать модули BPD, Info и Random, строка будет выглядеть так:

```
plugins = bpd rdm info
```

После активации модуля в строке

```
plugins =
```

в файле **~/.beetsconfig** вам станет доступна его справка, которую можно получить командой

```
beet ? pluginname
```

Так, для получения справки про модуль random скомандуйте

```
beet ? Random
```

LXF

Скорая помощь

Beets работает только с *MusicBrainz*, и если исполнителя/альбома там нет, *Beets* не получит никаких метаданных. Подумайте о том, чтобы самим добавить альбом в *MusicBrainz*.

ownCloud:

Марко Фиоретти рассказывает о свободной альтернативе Dropbox, Flickr и многих других онлайн-сервисов.

Наш эксперт

Марко Фиоретти – автор «Семейного руководства по цифровой свободе», а также активист свободного ПО и программист.

Облачные вычисления – современное модное словцо для «использования удаленного компьютера и ПО через Интернет для решения своих задач вместо своего собственного». Gmail, Dropbox, Flickr, Tumblr и персональные блоги *Blogspot* или *WordPress* – лишь некоторые из ныне популярных облачных сервисов.

При всем своем удобстве, у облачных вычислений – по крайней мере, в перечисленных инкарнациях – есть большая проблема. С одной стороны, вы платите (персональными данными, вниманием и деньгами) кому-то за управление устройствами и программами от вашего имени. С другой стороны, все эти сервисы и ваш собственный контент остается доступным в Сети только до тех пор, пока провайдер остается доступным и продолжает пускать вас к себе на своих условиях.

Решение этой проблемы – создать собственное облако: портативное web-окружение на основе свободного ПО, которое можно запустить практически везде, от простых учетных записей на web-хостинге (включая ряд бесплатных) до вашего собственного (виртуального) сервера на Linux. На нашем уроке мы объясним, как пользоваться *ownCloud* (http://owncloud.org), одним из наиболее перспективных свободных проектов в этой сфере.

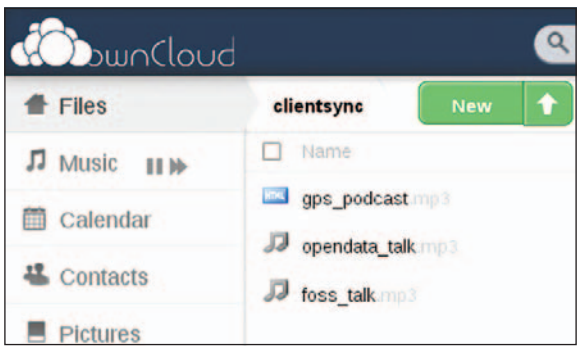


Рис. 1. Базовая версия *ownCloud*: хранение файлов, прослушивание музыки онлайн, галереи изображений, календарь и адресные книги — все в одном и том же окне браузера.

Чтобы раздразнить ваш аппетит и повысить мотивацию, руководство разделено на две основные части. В первой представлено, что конечные пользователи могут делать с уже установленной *ownCloud*. Во второй показано, как установить, настроить и администрировать ее.

Базовые сервисы *ownCloud* представляют собой смесь онлайн-хранилища файлов, календарей и адресной книги. Обращаться к папкам в *ownCloud* и синхронизировать их можно прямо с рабочего стола. С одной и той же установленной программой может работать много независимых пользователей.

Начиная с версии 4.5 (описанной на этих страницах), ownCloud поддерживает контроль версий файлов и унифицированный интерфейс доступа и синхронизации с крупнейшими сервисами хранения данных, такими как Dropbox или Google Drive. Если этого мало, возможности программы можно расширить несколькими дополнительными приложениями.

Базовая версия *ownCloud* выглядит так, как показано на рис. 1: место для хранения и организации общего доступа к вашим файлам, изображениям, аудиофайлам, календарю и к списку контактов из любого браузера с любого компьютера или смартфона. В файловом менеджере есть средства просмотра форматов OpenDocument и PDF, а также простой редактор текстовых файлов. Изображения доступны для просмотра по отдельности или в полноэкранном слайд-шоу, а для коллекции музыки доступно потоковое воспроизведение.

Функции контактов и календаря выглядят не слишком современно, но обладают всеми основными функциями. Можно импортировать контакты из Google или создать произвольные поля в адресной книге или в календаре. Можно задать несколько независимых календарей и поделиться ими с другими пользователями в одном и том же *ownCloud*. При первом запуске календаря он спросит о вашем местоположении для выбора часового пояса, но его можно изменить. Помните, что если задать его неправильно, календарь в *ownCloud* не сможет синхронизироваться с календарями на смартфоне и на настольном компьютере.

Даже если автоматическая синхронизация невозможна, можно импортировать и экспортировать события или целые календари из вашей учетной записи *ownCloud* в стандартном формате iCal. С этим форматом сегодня работают все приличные программы для работы с календарем, включая те, что для смартфонов.

Контейнер приложений

Ко всему хорошему, у *ownCloud* есть еще одна функция – пока малоиспользуемая, но, по нашему скромному мнению, имеющая огромный потенциал: кажется, *ownCloud* родилась контейнером и унифицированной точкой доступа для совершенно независимых сторонних онлайн-сервисов на базе открытого ПО. Администратор *ownCloud* может расширить эти сервисы разными способами, установив множество дополнительных приложений, список которых доступен на http://apps.owncloud.com.

Личное облако

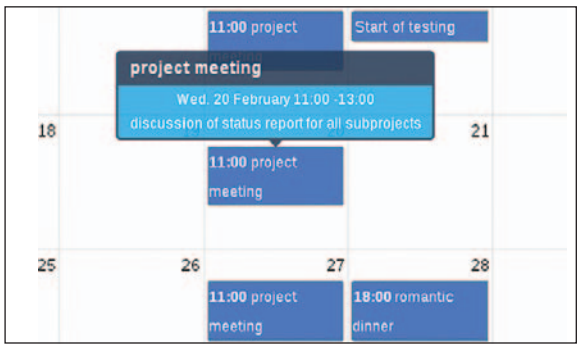


Рис. 2. Календарь *ownCloud*: прост, но удобен, и легко синхронизируется с настольным компьютером или мобильными устройствами.

А велики ли ваши файлы?

По умолчанию ownCloud ограничивает объем одного загружаемого (по крайней мере, через web-интерфейс) файла 512 МБ, и этого, разумеется, слишком мало для многих пользователей. Этой теме посвящена целая страница в официальной документации, но вкратце – можно легко увеличить этот размер до того, который позволяет ваш провайдер, если у вас не собственный сервер. Для этого достаточно добавить раздел

```
<IfModule mod_php5.c>
php_value upload_max_filesize 4096M
php_value post_max_size 4096M
</IfModule>
```

в конфигурационный файл PHP или в отдельный файл. Однако это сработает, только если администратор разрешил такое делать в настройках web-сервера и интерпретатора PHP! Подробности см. на http://owncloud.org/support/big-files.

Мы представим небольшую подборку таких приложений, просто чтобы пояснить общую идею. Например, хранить заметки на *ownCloud* можно по крайней мере в трех разных программах. Самая простая и проще всего синхронизируемая с мобильными устройствами – *Notes*: в ней можно хранить описания задач, списки дел и другие подобные вещи в виде простых текстовых файлов. *AtNotes* поддерживает базовое форматирование HTML и тэги для заметок. *ownCloud Journal* идет на шаг дальше – в нем можно сортировать, фильтровать и искать по диапазону даты или времени все дневниковые записи, хранимые в нем. Внутри него записи журнала сохраняются как записи календаря *ownCloud*, и ими можно делиться с другими пользователями.

Приложения *ownCloud* могут не только хранить заметки. *Theme Manager* позволит администратору менять внешний вид всего сайта несколькими щелчками мыши. Есть проигрыватель *impress.js* для генерации слайд-шоу и приложение для синхронизации настроек *Firefox* по всем компьютерам и учетным записям. Есть интерфейс для публикации загруженных в *ownCloud* фотографий в блогах *WordPress*. Мы также думаем, что многие пользователи полюбят *Bookmark Manager* с рис. 3 или *Shorty*, приложение *ownCloud*, которое упорядочивает закладки и дает ими поделиться с помощью нескольких внешних сервисов сокращения URL-адресов.

Но будущее ownCloud, возможно, в интеграции: зачем заново изобретать все велосипеды, когда можно собрать их из готовых запчастей? На рис. 4 показан уже работающий пример этой категории приложений *ownCloud*: оно позволяет вам зайти в вашу учетную запись *Roundcube* и использовать ее как вкладку *ownCloud*, вместо разработки еще одного бесплатного почтового клиента с web-интерфейсом! Разве это не прекрасно?

Конечно, все это требует небольшой настройки на стороне пользователя. Первое, что нужно сделать с новой учетной записью *ownCloud* – щелкнуть по шестеренке в левом нижнем углу. Откроется панель настройки пользователя, где вы найдете набор важнейших параметров или выполните важные операции. Параметры, которые можно (нужно!) настроить, включают адрес электронной почты для восстановления пароля и язык пользовательского интерфейса. Да, и не забудьте скопировать значение URL-адреса WebDAV. Если во время установки было включено управление версиями, вы также можете отменить старые версии файлов, которые вам больше не нужны. На конфигурационной панели каждый пользователь может загрузить в виде одного zip-архива все свои файлы и данные в формате, который можно одним щелчком загрузить в другом экземпляре *ownCloud*. Попробуйте-ка сделать это на закрытых облачных сервисах!

ownCloud с рабочего стола

Доступ с рабочего стола возможен в общем из любого файлового менеджера с поддержкой протокола WebDAV: в него нужно просто ввести URL-адрес WebDAV, указанный в конфигурационной панели. Подробные инструкции по использованию таких URL с несколькими файловыми менеджерами доступны на http://owncloud.org/support/webdav.

Кроме файловых менеджеров, есть и настольные клиенты *ownCloud* для постоянной синхронизации локального каталога на компьютере с онлайн-аккаунтом, почти так же, как в *Dropbox*. На январь 2013 года на сайте были клиенты для Linux, Windows (XP, Vista, 7 и 8, 32- или 64-битные версии) и Mac OS X 10.5 или выше, а также для 64-битной архитектуры Intel.

Установить клиент *ownCloud* в Linux просто: на сайте есть ссылки на репозитории с двоичными пакетами для большинства популярных дистрибутивов. Мы загрузили версию 1.1.4 клиента для компьютера x86_64 с Fedora 17, но не смогли им воспользоваться. Настройка была очень простой, и синхронизация работала так, как ожидалось... но лишь несколько часов. Согласно лог-файлу клиента, причина была в том, что “Csync failed to load the state db [Csync не удалось загрузить базу данных состояния]”: проблема, которая судя по форумам *ownCloud*, должна быть решена еще с версии 1.1.2. Надеюсь, она будет решена, когда вы будете читать это.

Установить *ownCloud* и работать с ним без особых проблем сможет любой, у кого есть базовые знания LAMP (Linux, *Apache*, *MySQL*, PHP): по существу, если вы уже устанавливали *WordPress* или другие подобные пакеты, у вас есть все навыки для управления *ownCloud*. А если нет, не беспокойтесь: это гораздо проще, чем кажется.

Перед загрузкой *ownCloud* нужно кое-что проверить. Во-первых, web-пространство, в котором будет жить *ownCloud*. Сам *ownCloud* занимает очень мало места, но если вы хотите использовать »

» **Не хотите пропустить номер?** Подпишитесь на www.linuxformat.ru/subscribe/!

его в качестве онлайн-хранилища, возможно, лучше сразу выбрать учетную запись с достаточным пространством. Далее, если только вы не абсолютно уверены, что вам не понадобятся постоянные web-адреса ни для чего помещаемого на облаке, нужно купить доменное имя. Разумеется, для правильной работы самого *ownCloud* оно совсем не требуется – просто при переносе установки на другой сервер с другим доменным именем все старые адреса – ссылки на галереи изображений и т.д. – перестанут работать. Это может быть очень плохо или почти не важно в зависимости от ваших потребностей; решать вам.

Мы также настоятельно рекомендуем проверить все компоненты *ownCloud* по крайней мере в течение недели после первой установки, прежде чем выдавать учетные записи другим или публиковать адреса файлов, которыми вы хотите поделиться. Тогда вы сможете безопасно удалить и переустановить все с другой конфигурацией, прежде чем работать в полноценном режиме.

Установка *ownCloud* и наши советы были опробованы на виртуальном частном сервере с Centos 6.3 и *Apache*. Инструкции для других дистрибутивов можно найти на http://owncloud.org/support/distro-notes, но почти все, о чем мы будем говорить, применимо как есть к любой учетной записи web-хостинга, который поддерживает PHP и базы данных *MySQL*, *PostgreSQL* или *SQLite*. Основное исключение – настройки web-сервера и PHP. На сайте *ownCloud* есть хорошая документация для нескольких web-серверов, не только для *Apache*. В любом случае спросите у провайдера, какими серверами он пользуется, и, самое главное, есть ли у них непонятные еще не задокументированные настройки PHP, которые могут усложнить настройку *ownCloud* (поверьте: такое бывает!).

Выбор базы данных

Последнее, что нужно сделать перед установкой ownCloud – выбрать, какой базой данных он будет пользоваться: *SQLite*, *MySQL* или *PostgreSQL*? С первой проще работать и создавать резервные копии: база данных *SQLite* – это один большой файл, который PHP может полностью обрабатывать сам, если установлены нужные библиотеки.

Если вы выберете *SQLite*, *ownCloud* создаст файл базы данных явно, в том же каталоге, где хранятся ваши файлы. Для двух других вариантов понадобятся база данных и учетная запись на отдельном сервере *MySQL* или *PostgreSQL*. Это не проблема, так как многие учетные записи web-хостинга включают одну базу данных *MySQL* или позволяют добавить ее за несколько фунтов в год дополнительно. В документации по *ownCloud* рекомендуется выбирать *MySQL* или *PostgreSQL*, если пользователей много и/или если они работают одновременно.

В некоторых дистрибутивах GNU/Linux (самый свежий список можно найти на вышеупомянутых сайтах) есть двоичные пакеты *ownCloud*, с которыми установить и обновлять программу – пара пустяков. Однако для *ownCloud* не обязательно нужен Linux, и главная причина пользоваться им – портируемость. Поэтому в нашем руководстве мы сознательно обойдем эти случаи и объясним, как скомпилировать программу из исходных кодов, которые можно загрузить с сайта.

› Рис. 3. Собственные менеджеры закладок *ownCloud* не так сложны, как другие подобные программы, но очень удобны.

Процедура установки *ownCloud* похожа на процедуру установки *WordPress*, *Drupal* и многих других свободных пакетов из LAMP (Linux/*Apache*/*MySQL*/PHP). Сначала нужно поместить некоторые файлы в определенный каталог web-сервера, затем настроить базу данных и другие параметры из браузера. У сервера должны быть права записи на этот каталог, в противном случае *ownCloud* не сможет создать некоторые каталоги, нужные ему для работы. На сайте *ownCloud* есть мастер, пользоваться которым, возможно, не стоит, хотя он и немного упрощает помещение файлов в каталог web-сервера. Чтобы запустить его, нужно загрузить (назовем наш сайт myowncloud.example.com) небольшой PHP-файл **setup-owncloud.php** в будущий каталог *ownCloud* и затем открыть в браузере myowncloud.example.com/setup-owncloud.php.

На этой странице можно проверить, совместима ли конфигурация PHP и сервера с *ownCloud*, и, нажав кнопку, загрузить настоящий код и установить его в тот же каталог. После этого установка продолжится как описано ранее, то есть как если бы вы устанавливали файлы вручную. Проблема с мастером только в том, что с ним нельзя установить текущую версию *ownCloud*. Версия мастера на начало января 2013 года оставит вас с *ownCloud 4.0* вместо *4.5*. Поэтому мы рекомендуем другой вариант установки: просто загрузите архив с последней стабильной версией (менее 10 МБ для *ownCloud 4.5*) в каталог на сервере и распакуйте его там.

На данном этапе, если вы не планируете использовать *SQLite*, убедитесь, что вы можете воспользоваться или создать, если ее нет, базу данных и пользователя *MySQL* и *PostgreSQL* специально для *ownCloud*. Вот команды, которыми это можно сделать с нуля в *MySQL*, самой распространенной базе данных на простых web-хостингах:

```
mysql> CREATE USER 'oc_user'@'localhost' IDENTIFIED BY
'the_mysql_password';
mysql> CREATE database owncloud_test;
mysql> GRANT ALL PRIVILEGES ON owncloud_test.* TO 'oc_
user'@'localhost' IDENTIFIED BY 'the_mysql_password'
```

Эти команды или их эквиваленты можно выполнять из любого доступного интерфейса *MySQL* на сервере, будь то web-утилита вроде *PHPMyAdmin* или клиент командной строки для Linux. Покончив с базой данных, откройте в браузере myowncloud.example.com. Вы увидите простую форму, которая запросит у вас имя пользователя и пароль администратора, абсолютный путь до корневого каталога *ownCloud* (не забудьте, что у сервера должны быть права записи для него!), тип базы данных и, если у вас не *SQLite*, ее расположение, имя пользователя и пароль. Это все, что касается установки!

Общие настройки

Помните ту маленькую иконку шестеренки в левом нижнем углу? Если щелкнуть по ней, зайдя в систему как администратор, на ней также появится вкладка Admin [Администратор]. На ней можно создавать учетные записи для других пользователей и добавлять их в группы; у каждой группы есть свой администратор. Группы нужны, если вы хотите или вам нужно максимально гибко контролировать, чем пользователи могут делиться друг с другом и с остальным Интернетом. Кроме регулирования пользователей в группах, у каждого из них может быть собственная квота дискового пространства. Даже если вы единственный пользователь своего *ownCloud*, создайте другого пользователя для выполнения повседневных задач. Учетная запись администратора должна применяться только для администрирования, хотя бы только для того, чтобы вам было сложнее испортить что-то по ошибке! Кроме управления пользователями, на административной панели *ownCloud* можно загружать и экспортировать основные

» **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

системные файлы, пользовательские файлы или все облако (пользовательские данные, файлы и базу данных) для резервного копирования или миграции. Хотя это делается одним щелчком, щелкать нужно вручную.

Установить приложения, найденные на apps.owncloud.com или еще где-нибудь, просто: загрузите и распакуйте архивы с исходными кодами в подкаталог с приложениями *ownCloud*, затем настройте их в разделе административной панели с приложениями.

Обновления и апгрейды

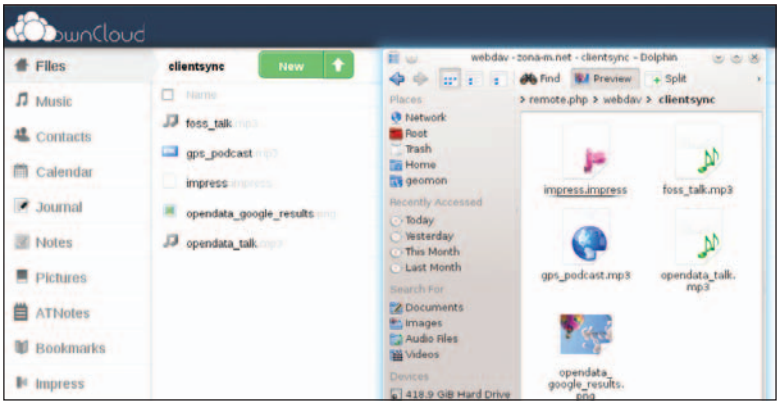
В *ownCloud* четко различаются обновления [update] и апгрейды [upgrade]. Обе процедуры просты и доходчиво объяснены на сайте *ownCloud*, поэтому мы отметим лишь самые важные моменты, которые нужно знать заранее. Первый момент очевиден: обязательно создавайте резервную копию всей системы перед любыми подобными изменениями!

Обновление – это изменение с одного релиза на другой, например, с версии 4.5 на 4.6. А апгрейд, по определению, переход на версию с бо́льшим номером. Это не просто формальное определение без практических последствий, так как обновления выполняются последовательно с последней минорной версии в качестве стартовой точки.

Другими словами: хотя пропустить промежуточные версии *ownCloud* – не катастрофа, это довольно надежный способ наткнуться на интересные, обычно недокументированные пакости. Чаще всего источник таких проблем одинаков у всех программ, состоящих из «PHP и базы данных». Обновление такой программы почти всегда означает и установку новых файлов PHP, и запуск в фоне неких автоматизированных *SQL*-запросов для обновления структуры и содержимого базы данных.

Затоптанные по ошибке файлы легко найти и восстановить. Дать волю *SQL*-запросам на базе данных, с не совсем той структурой, для которой они предназначались, гораздо хуже. Они могут повредить базу данных так, что останется только полностью переустановить программу из резервной копии предыдущей версии. Поэтому сделайте себе одолжение: никогда не пропускайте выход новых версий и постоянно поддерживайте *ownCloud* максимально обновленным.

Еще одна проблема с обновлениями, которую нельзя игнорировать – в том, что происходит с дополнительными приложениями или с приложениями сторонних разработчиков при каждом изменении *ownCloud*. С этой точки зрения обновления никогда не должны стать проблемой, но для апгрейдов потребуется чуть больше усилий. В общем случае перед выполнением апгрейда нужно дезактивировать все приложения сторонних разработчиков на административной панели. Кроме того, может случиться так, что используемое приложение *ownCloud* несовместимо с новой старшей версией. Тогда нужно или отложить апгрейд, или временно отключить это приложение.



› Рис. 4. Где этот файл? В *ownCloud* или на жестком диске? Неважно — можно обратиться к нему обоими способами!

Основное достоинство *ownCloud* – переносимость. Правильно настройте его один раз (особенно с собственным доменным именем!), и вы сможете воссоздать его снова в любой момент с минимальными усилиями, на том же или любом другом web-сервере, безо всяких проблем. Чтобы это было возможным, нужно настроить автоматическое, регулярное резервное копирование по меньшей мере трех вещей. Первая – каталог с данными, который содержит все пользовательские файлы и другие данные. Вторая – база данных. Если вы пользуетесь *SQLite*, этот шаг входит в предыдущий. Если нет, нужно настроить ежедневную задачу *cron*, которая будет экспортировать всю базу данных в один текстовый файл, который вы добавите в резервную копию с остальными. Экспорт можно выполнить командой **mysqldump** или, в *PostgreSQL*, **pg_dump**.

Следующее, что нужно защитить – файл **config.php** в подкаталоге **config**, который содержит имя пользователя и пароль для подключения к базе данных и другие глобальные переменные. В зависимости от сервера, также может потребоваться резервная копия конфигурационных файлов HTTP и/или PHP. Если вы делаете это регулярно, например, с ежедневным заданием *cron*, перенос вашего *ownCloud* на другой сервер или восстановление после сбоя должны пройти почти безболезненно.

Сначала убедитесь, что конфигурация web-сервера/PHP нового сайта идентична или, по меньшей мере, совместима с конфигурацией старого сайта. Затем распакуйте первоначальную версию исходников той же версии *ownCloud*, которая стоит у вас (у вас обязательно должна быть и копия исходного архива!), и воссоздайте базу данных.

Наконец, поместите каталог с данными и файл **config.php** на их место в корневом каталоге нового *ownCloud*. Если имя пользователя и пароль для внешней базы данных должны отличаться от исходных (у некоторых провайдеров собственные правила создания имен и паролей...), запишите новые значения в **config.php**. Вот и все, правда! LXF

А что у нас с безопасностью?

Безопасность в *ownCloud* заслуживает отдельного раздела по двум причинам. Во-первых, это очень важно, поэтому должно выделяться, хотя бы визуально. У нас недостаточно места, чтобы рассказать об этом исчерпывающе, и эта информация в любом случае не будет такой актуальной, как на сайте, но мы хотим, чтобы вы знали: о безопасности надо обязательно думать.

Еще одна причина коснуться темы лишь вкратце в том, что по сути сказать надо мало. Основные действия, выполняемые для защиты *ownCloud*, почти те же, что и для любого приложения LAMP. Они очень хорошо описаны в Интернете, поэтому мы лишь упомянем их, чтобы вы знали, что искать.

Для начала включите базовую аутентификацию на своем web-сервере. В *Apache* нужно добавить примерно такую запись в файл конфигурации, чтобы включить файл управления доступом, который есть в *ownCloud*:

```
<Directory /complete/path/to/owncloud/>
AllowOverride All
</Directory>
```

Это просто пример. Зайдите на сайт *ownCloud*, чтобы найти подробную, актуальную информацию об аутентификации и директивах HTTPD для своего сервера и своего дистрибутива Linux.

Второе – нужно получить сертификат SSL для защищенных HTTP-соединений. Последнее, но не по важности – обеспечьте, чтобы неавторизованные пользователи в вашей системе не могли обращаться к одиночным файлам в *ownCloud*, последовав указаниям на форуме http://forum.owncloud.org/viewtopic.php?f=3&t=6659.

ЧАСТЬ 4

Введение

Джоно Бэкон закруляет серию несколькими советами профессионала.

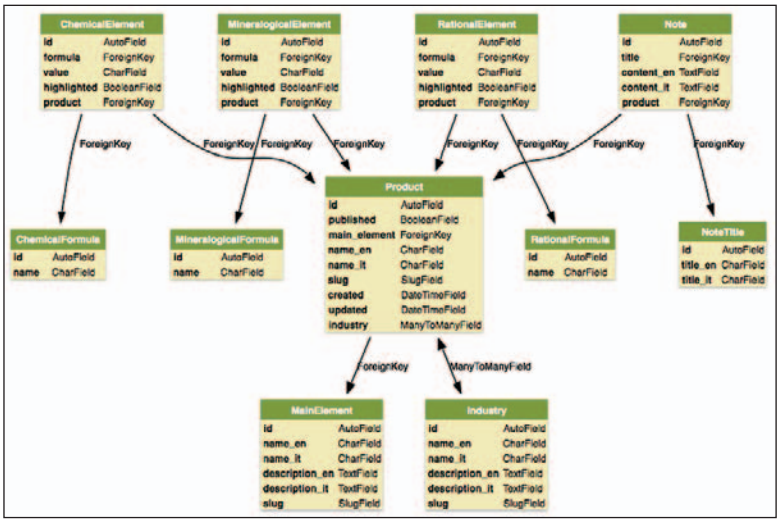
Наш эксперт

Джоно Бэкон управляет сообществом Ubuntu; он автор книги «Искусство сообщества» и основатель ежегодного саммита руководителей сообщества.

В предыдущих трех частях этой серии о знакомстве с Django мы открыли несколько основных принципов создания мощного динамического web-приложения. Мы поговорили о настройке, базах данных, моделях, формах, задании URL-адресов, шаблонах, темах и др.

В этой четвертой и последней статье мы заполним ряд оставшихся в материалах пробелов и покопаемся в приемах и советах, способных вам пригодиться в повседневном программировании с Django. Эта статья предполагает наличие базовых знаний во многих из этих областей, и в ней я покажу вам некоторые решения ряда распространенных задач, которых в этой серии мы еще не касались.

› Рис. 1. На схеме удобно представлены взаимосвязанные отношения.



Новые трюки шаблонов

В предыдущей статье мы узнали, как построить интерфейс web-страницы с помощью шаблонов Django. Основы мы рассмотрели, но прочные знания языка шаблонов Django помогут вам создать лучший, более настраиваемый интерфейс.

При написании шаблонов важно хорошо знать два основных типа выражений: циклы и условия. Мы уже рассматривали циклы с циклом **for**:

```
{% for i in content %}
  {{ i }}
{% endfor %}
```

При отправке данных из представления шаблону обычно отправляется набор элементов, сгенерированный запросом, например, так:

```
content = MyTable.objects.all()
return render_to_response('mytemplate.html', {'content': content},
context_instance=RequestContext(request))
```

Тут по ним можно пройтись в цикле **for** с приведенным выше синтаксисом. Иногда нам нужен больший контроль над отправляемыми шаблону данными. Скажем, в разных программах на Python структуры данных представляются по-разному. Если запрос подобен приведенному выше, вы получаете список элементов, по которому легко пройтись в цикле. Но иногда может понадобиться воспользоваться словарем – например, так:

```
{ "one" : 1, "two" : 2, "three" : 3 }
```

Тогда данные генерируются в представлении и затем отправляются странице точно таким же образом. Простой пример:

```
content = [ "one" : 1, "two" : 2, "three" : 3]
return render_to_response('mytemplate.html', {'content': content},
context_instance=RequestContext(request))
```

Итерировать такие данные в шаблоне нужно немного иначе. Для этого воспользуемся следующим форматом:

```
{% for k, v in content.items %}
  <p>The key is {{ k }} and the value is {{ v}}</p>
{% endfor %}
```

Здесь примечательны несколько моментов. Во-первых, мы используем конструкцию **.items** после указания массива content, по которому идут итерации – так мы получаем компоненты структуры данных. Затем мы получаем два компонента, поскольку это словарь, и **k** ссылается на ключ, а **v** – на значение. Можно сделать еще умнее – сформировать список словарей следующим образом:

```
content = [{ "one" : 1, "two" : 2 }, { "three" : 3 }, {"four" : 4 }]
```

Для итераций по нему нужно сначала создать цикл по внешнему списку, а внутри него выбирать каждый словарь и получать значения:

```
{% for item in content %}
  {% for k, v in item.items %}
    <p>The key is {{ k }} and the value is {{ v }}</p>
```

в Django

```
  {% endfor %}
{% endfor %}
```

Еще один распространенный пример – словарь из словарей, например, при построении набора данных, которые относятся к чему-то. Например:

```
content = {
  “bob” : { “job” : “Programmer”, “id” : 234},
  “jane” : { “job” : “QA”, “id” : 245 },
}
```

Здесь у нас есть словарь для каждого лица, а в словаре – всякая информация об этом лице. Обратите внимание, что в каждом словаре используется один и тот же ключ (здесь – **job**). Позже мы посмотрим, как использовать операторы условия в таких структурах, а пока просто проитерируем его:

```
{% for name, info in content.items %}
  <h1>{{ name }}</h1>
  {% for k, v in info.items %}
    <p><strong>{{ k }}</strong> : {{v }}</p>
  {% endfor %}
{% endfor %}
```

У нас есть внешний цикл **for**, который проходит по именам и получает данные в переменных **name** и **info**, и внутренний цикл, который проходит по словарям из **info**.

Другая распространенная ситуация, в которой вы однажды окажетесь – создание структуры данных, объединяющей вместе данные моделей Django и другие данные. Пусть, например, у нас есть таблица People, и мы хотим добавить дополнительные данные в структуру данных. В представлении они создаются так:

```
people = People.objects.all()
content = {}
for person in people:
  if person.clearance == 1:
    content[person] = { “здание” : “штаб”, “знаетсекреты” : True }
 else:
    content[person] = { “здание” : “офис”, “знаетсекреты” : False }
    return render_to_response(‘mytemplate.html’, {‘content’:
content}, context_instance=RequestContext(request))
```

Здесь мы используем объект данных как ключ внешнего словаря, а значение – наш словарь. Скажем, если в People есть запись “Bob” и у Боба есть допуск, наша структура данных будет выглядеть так:

```
{ <People: Bob> : { “здание” : “штаб”, “знаетсекреты” : True } }
```

Итерироваться по ней можно так же, как в предыдущем примере, но можно обращаться к полям объекта данных. Например:

```
{% for name, info in content.items %}
  <h1>{{ name.fullname }} has clearance level {{name.clearance }}
</h1>
    {% for k, v in info.items %}
<p><strong>{{ k }}</strong> : {{v }}</p>
{% endfor %}
{% endfor %}
```

Эти примеры должны покрыть большинство ситуаций при итерировании данных различных типов в своих шаблонах.

Памятка по фильтрам шаблонов

{% spaceless %}	Удаляет пробелы между тэгами HTML
{% ssi ‘/home/jono.foo.html’ %}	Включает в страницу содержимое указанного файла
{{ value|addslashes }}	Добавляет слэши перед кавычками
{{ value|capfirst }}	Делает первый символ заглавным
{{ value|cut:” “ }}	Удаляет все вхождения заданного символа
{{ value|default:”nothing” }}	Если значение по умолчанию ложно, использовать указанное значение
{{ value|filesizeformat }}	Форматирует размер файла в формат, удобный для восприятия человеком (например, 2,1 МБ)
{{ value|floatformat }}	Изменяет формат отображения чисел с плавающей точкой (можно применять дополнительное форматирование)
{{ value|length_is:”4” }}	Возвращает true, если значение value имеет указанную длину
{{ value|linenumbers }}	Отображает значение value с номерами строк
{{ value|lower }}	Преобразует строку в нижний регистр
{{ value|time:”H:i” }}	Форматирует время в заданном формате (в данном случае, 01.30)

Полный список фильтров см. в документации на https://docs.djangoproject.com/en/dev/ref/templates/built-ins.

Условия и фильтры

Теперь переключим передачу и рассмотрим условия в фильтрах. Условия используются для проверки соответствия определенных переменных заданным значениям, и в зависимости от результата проверки можно отображать различную информацию.

Пусть, например, у нас в представлении имеется такая структура данных:

```
content = [ “admin”, “bob”, “jane” ]
```

Это простой список имен. Теперь в нашем шаблоне можно воспользоваться командой **if**, чтобы проверить, есть ли в списке администратор, и вывести для него другое сообщение:

```
{% for name in content %}
  {% if name == “admin” %}
    <p>Admin logged in</p>
  {% else %}
    <p>Normal user logged in</p>
  {% endif %}
{% endfor %}
```

Другой способ это сделать – воспользоваться командой **ifequal** [если равно]:

```
{% for name in content %}
  {% ifequal name “admin” %}
    <p>Admin logged in</p>
```

»

» **Не хотите пропустить номер?** Подпишитесь на www.linuxformat.ru/subscribe/!

```
  {% else %}
    <p>Normal user logged in</p>
  {% endifequal %}
{% endfor %}
```

Еще одна приятная возможность шаблонов Django – широкий диапазон фильтров. Например, часто приходится отображать в таблице строки из базы данных, но если строк нет, нужно вывести соответствующее сообщение.

Для этого удобно воспользоваться фильтром length. Например, представьте, что нам нужно вывести список пользователей из таблицы People. Вот как можно проверить, есть ли в таблице записи, и вывести соответствующий результат:

```
{% ifequal contentlength 0 %}
  <p>No people.</p>
{% else %}
  {% for name in content %}
    {% ifequal name “admin” %}
      <p>Admin logged in</p>
    {% else %}
      <p>Normal user logged in</p>
    {% endifequal %}
  {% endfor %}
{% endifequal %}
```

Здесь мы пользуемся обычным выражением **ifequal**, но применяем фильтр **length** к переменной content с помощью символа |.

В Django есть несколько полезных фильтров. Рассмотрим несколько самых популярных. Прежде всего, вам может понадобиться отображать даты в различных форматах – например, определить локаль пользователя и представить дату в соответствующем формате (например, 1-11-2013 вместо 11-1-2013). К объекту **datetime** можно применить фильтр **date**, например, так:

```
{{ value|date:”D d M Y” }}
```

Он отформатирует дату следующим образом: Tue 08 Feb 2013. Буквы в этой строке определяют форматирование даты, и есть множество различных опций для того, чтобы представить дату именно так, как вам нужно. Еще два полезных фильтра – **dictsort** и **dictsortreversed**, он сортирует словарь, к которому применяется. Другие два – **first** и **last**, они возвращают первый и последний элементы списка. Некоторые полезные фильтры приведены во врезке «Памятка по фильтрам шаблонов».

Добавление своих команд

В процессе работы с Django вы очень близко познакомитесь со скриптом **manage.py**, входящим в ваш проект. Чаще всего он используется для выполнения команд таким образом:

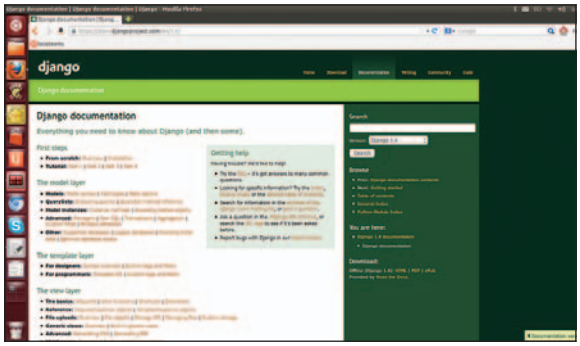
```
python manage.py syncdb
python manage.py runserver
```

Миграция с South

Некоторые из вас может слегка беспокоить, что произойдет, когда вы выпустите свое web-приложение Django, а затем захотите изменить структуру своих моделей. Что станет со всеми данными при изменении структуры базы данных? К счастью, есть прекрасное решение этой проблемы – South.

South – проект, который подключается к Django и позволяет изменить **models.py** и затем создать файл миграции. Этот файл сконвертирует базу данных согласно новым моделям и проведет перенос данных. Это безопасный способ изменения структуры базы данных.

Именно так мы рекомендуем управлять своей базой данных, когда у вас уже готов код приложения. Более подробно узнать о South можно на сайте http://south.aeracode.org.



› У Django фантастическая документация для изучения различных частей платформы.

manage.py – швейцарский армейский нож в арсенале Django. Он содержит команды для выполнения всевозможных действий: создания приложений, тестов, проверки моделей, исследования базы данных и др. Но иногда может возникнуть необходимость в собственных командах, которые можно выполнить с **manage.py,** например, когда нужно воспользоваться в своем проекте на Django функциями из другого проекта, не на Django.

Например, не так давно я начал проект «Достижения в Ubuntu [Ubuntu Accomplishments]» – это система, которая фиксирует ваши достижения и награждает вас призами за них. В рамках этого проекта я написал сервер проверки, который проверяет призы на подлинность для защиты от пользователей, подделывающих свои призы.

Потратив некоторое время на написание сервера проверки, я понял, что мне нужен способ визуализации ошибок и проблем, обнаруженных сервером. В качестве такого способа я выбрал клиент на Django и принялся за работу.

В процессе работы я столкнулся с проблемой: когда сервер проверки обнаружит ошибку – как добавить запись в базу данных Django, чтобы ее увидел клиент на Django? С одной стороны, можно было бы обновить базу данных вручную, подключившись к ней напрямую, но это неряшливый и штукарский подход. Затем я узнал, что можно создать пользовательскую команду для **manage.py**, которая бы это делала. Вот я и создал команду **addfailure**, и сервер проверки просто вызывал ее при возникновении ошибки и передавал команде информацию об ошибке. Посмотрим, как создать такую команду.

Эти различные команды **manage.py** связываются с приложением. Можно создать набор команд для каждого приложения в проекте. Например, предположим, что у нас есть проект Django для хранения списка сотрудников компании. В этом проекте есть модель ‘Staff [Штат]’, которая хранит список сотрудников. Предположим также, что приложение называется ‘company [компания]’.

Управление

Пользовательскую функциональность, относящуюся к скрипту **manage.py**, назовем Management [Управлением]. Для начала нужно создать каталог **management** внутри нашего приложения, а затем, внутри этого каталога, каталог **commands**, где будут храниться наши команды (в данном случае, у нас получилось **company/management/commands**).

Теперь создадим пользовательскую команду и назовем ее **addstaffmember** [Добавить сотрудника]. Для этого создайте файл **addstaffmember.py** в каталоге **commands**. В этом каталоге также нужно создать пустой файл **__init__.py**. Теперь откроем **addstaffmember.py** и добавим в него немного кода.

» **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

Сначала нужно загрузить некоторые модули, которыми мы воспользуемся для реализации команды:

```
from django.core.management.base import BaseCommand,
CommandError
```

Теперь создадим новый класс Command типа BaseCommand и добавим в него функцию handle, в которой будет хранится логика команды:

```
class Command(BaseCommand):
  def handle(self, *args, **options):
    print “hello world”
```

Сохраните файл. К счастью, Django довольно умен и автоматически регистрирует все команды из каталога **commands**, которые не начинаются с символа подчеркивания. Поэтому, запустив команду **python manage.py**, вы должны увидеть и свою команду в списке. Для полноты картины добавим короткую строку с описанием того, что делает команда. Эта строка появится, когда кто-то запустит **python manage.py help addstaffmember**. Для этого добавьте строку **help** после определения класса:

```
class Command(BaseCommand):
  help = ‘добавить сотрудника в базу данных’
```

В текущем виде наша команда не особо полезна. Нам нужно научиться принимать наборы аргументов, которые мы сможем обработать и добавить в базу данных. Сначала определим аргументы, которые мы принимаем, для чего зададим переменную **args** точно так же, как ранее задавали переменную **help**. Внутри переменной **args** мы укажем принимаемые аргументы. Например:

```
class Command(BaseCommand):
  args ‘firstname lastname job’
  help = ‘добавить сотрудника в базу данных’
```

Если теперь запустить **python manage.py help addstaffmember**, вы увидите строку **help** и список ожидаемых аргументов команды. Хотя мы определили эти аргументы, их имена не точно соответствуют переменным скрипта. Вместо этого нужно прочесть аргумент по его положению и записать его значение в переменную. Например:

```
item_firstname = args[0]
item_lastname = args[1]
item_job = args[2]
```

Мы считали свои данные, и можно добавить их в базу. Для начала загрузим нашу модель:

```
from company.models import Staff
```

Теперь можно создать запись **Staff**, в которую мы передаем данные переменных и затем сохраняем ее в базу данных:

```
entry = Staff.objects.create(firstname=item_firstname,
lastname=item_lastname, job=item_job)
entry.save()
```

Наша команда готова! Можно проверить ее так:

```
python manage.py testcommand “Alan” “Cox” “Kernel Programmer”
```

Когда мы запустим эту команду, наша таблица Staff обновится, и вы должны увидеть ее в своем проекте! Вот полный скрипт команды:

```
from django.core.management.base import BaseCommand,
CommandError
from company.models import Staff
class Command(BaseCommand):
  args = ‘firstname lastname job’
  help = ‘Adds a staff member to the database.’
  def handle(self, *args, **options):
    item_firstname = args[0]
    item_lastname = args[1]
    item_job = args[2]
      entry = Staff.objects.create(firstname=item_firstname,
lastname=item_lastname, ob=item_job)
      entry.save()
```

Понимание своих моделей

› Обязательно помечайте свои вопросы на stackoverflow.com тэгом Django.

Как и у большинства новичков в Django, в ваших первых проектах будет совсем немного моделей. По мере освоения системы, с ростом проектов от небольших до полноценных приложений, которыми должны будут пользоваться и другие, ваши модели будут расти и расширяться. Вскоре у вас будет огромный список взаимосвязанных моделей, и с изменением дизайна и функциональности будут меняться и ваши модели. В результате вы будете улучшать свои модели шаг за шагом в надежде создать окончательную версию дизайна, чтобы больше не изменять модели, когда проект начнет использоваться.

Перед окончательным релизом вы, вероятно, захотите полностью просмотреть свои модели, чтобы убедиться, что в них нет никаких бессмысленных связей. Подробно рассмотреть такое множество взаимосвязанных моделей сложно. Поэтому полезно иметь схему всех своих моделей и связей между ними. К счастью, сидеть и рисовать ее вручную не придется – ее можно сгенерировать автоматически.

Для этого нужно установить расширения Django (Django Extensions). Это набор полезных дополнений к функциональности **manage.py**. Для вашего дистрибутива должен быть соответствующий пакет (например, в Ubuntu я установил *python-django-extensions*). Исходный код можно загрузить напрямую с https://github.com/django-extensions/django-extensions.

Чтобы воспользоваться расширениями, нужно убедиться, что они добавлены как приложение в **settings.py** вашего проекта в раздел INSTALLED_APPS. Добавьте в INSTALLED_APPS следующее:

```
django_extensions
```

Для построения схемы нужно запустить команду, которая сгенерирует схему GraphViz; это специальный язык разметки, используемый для генерации схем. Обычно затем с помощью другой утилиты схема конвертируется в удобный для просмотра вид. И все это можно сделать одной командой!

```
python manage.py graph_models company -a -g -o my_project.png
```

Здесь мы выполняем команду **graph_models** для приложения company и генерируем файл **my_project.png** с нашей схемой. Пример такой схемы показан на рис. 1.

Django – фантастический каркас для создания мощных web-приложений с возможностями, расширяемостью и удовольствием от работы, сравнимыми со всем этим при написании приложений на Python. В нашей серии мы исследовали множество тем, но многие остались за ее рамками. Рекомендую погрузиться в превосходную документацию на https://docs.djangoproject.com/en – и обязательно задавать вопросы на сайтах http://stackoverflow.com, помечая их тэгом Django. Еще один прекрасный источник помощи – канал #django в IRC-сети Freenode.

Если вы следовали за нами в течение всей серии о Django, спасибо вам за это. Удачи с будущими проектами, и обязательно сообщите мне о своих успехах. Всего доброго! LXF

ЧАСТЬ 1

CUDA: Ускоряем

Константин Калгин и **Евгений Балдин** ухватились за игровые технологии с целью продвинуть решение научных задач.

Наш эксперт

Константин Калгин
Программист, который превращает персоналку в маленький супер-компьютер.

Наш эксперт

Евгений Балдин
Физик, который действительно знает, что такое нехватка вычислительных ресурсов.

Вам нужны гигафлопсы здесь и сейчас? Вы не хотите ждать светлого будущего OpenCL и согласны ради действительно стоящего дела вляпаться в «вендор-лок»? Тогда выбора как бы и нет – CUDA ждет нового адепта.

Предыстория

Как правило, люди тратят деньги либо на хлеб, либо на зрелища. На первое тратят потому, что хочется дожить до второго. Игровой плебс требует красивой картинки – так и возникли графические ускорители.

Изначально графические ускорители (видеокарты, GPU, Graphical Processing Units) предназначались для вывода двумерной или трехмерной графической информации на экран. Долгое время процесс отображения графической информации на экране управлялся лишь структурами данных – массивами примитивных фигур, текстурами и простейшими цветовыми фильтрами.

Только в XXI веке, хоть и в самом его начале, в графических ускорителях появилась поддержка шейдеров или мини-программ обработки данных на различных стадиях графического конвейера. Процесс отображения картинки теперь стал управляться не только данными, но и мини-программами, исполняющимися на самом графическом ускорителе. Шейдеры дали возможность разработчикам создавать свои собственные спецэффекты, а не ограничиваться уже встроенными в железо. Это значительно увеличило сложность и реалистичность компьютерной графики и, соответственно, привлекательность итоговой картинки для избалованного зрителя.

Фактически сразу после первых шейдеров графические ускорители начали использоваться энтузиастами для решения нерафических задач, то есть для задач общего назначения (GPGPU, General Purpose computations on GPU). Начали появляться статьи, в том числе и в научных журналах, о применении графических ускорителей для решения довольно важных и часто используемых задач линейной алгебры, а также задач, которые не имеют аналитического решения в общем виде – вроде моделирования системы N гравитирующих тел. Зачем серьезные, казалось бы, люди тратили силы и время на железо, целиком и полностью ориентированное на игровую аудиторию? Ответ прост: цена на производительность. В силу большого спроса казуальной аудитории на развлечения графические ускорители стали мощными и относительно дешевыми, хотя и узко специализированными вычислительными системами. На пути энтузиастов была только одна проблема: как бы сделать процедуру загрузки всей этой мощи попроще?

С появлением в 2007 году программно-аппаратной архитектуры CUDA графических ускорителей компании Nvidia ситуация кардинально изменилась. Программы стали составляться не на специальном языке описания шейдеров, а на знакомом C/C++. В описании графических ускорителей CUDA практически перестали использоваться графические термины, такие как шейдеры, точки, текстуры, фильтрация, Z-буфер и пр. Потенциально серьезный конкурент в лице OpenCL появился только в 2010 году, поэтому на сегодня CUDA является фактически единственной зрелой технологией использования графических ускорителей. Во множестве учебных центрах существуют обучающие программы по технологиям CUDA, и множество специалистов уже владеют необходимыми знаниями на вполне приемлемом уровне.

К недостаткам CUDA следует отнести закрытость драйверов и непереносимость кода за пределы платформ от Nvidia. Первое может приводить к весьма странным аберрациям в поведении пользовательских программ и необъяснимым падениям производительности в зависимости от версии компонент этой программно-аппаратной платформы. Второе не позволяет отказаться от нее в пользу использования других платформ, которые сейчас достаточно активно развиваются и в перспективе могут обогнать системы Nvidia.

Самый производительный суперкомпьютер в Top500 (http://www.top500.org) на ноябрь 2012 года включает в качестве одного из своих элементов ускорители Nvidia. Почти десять процентов систем из этого списка также указали ускорители Nvidia в описании своей архитектуры, и эта доля в обозримом будущем будет расти.

Установка CUDA

Предполагается, что в случае серьезной работы, например, на ближайшем доступном университетском кластере, все уже установлено. Если же хочется начать изучать технологию CUDA в домашних условиях, то необходимо для начала установить ее программную часть.

Естественно, необходимо убедиться, что в вашем компьютере есть графический ускоритель от компании Nvidia и что он поддерживает CUDA. За информацией можно обратиться к той же Википедии: http://ru.wikipedia.org/wiki/CUDA.

Затем нужно установить закрытый двоичный драйвер посвежее от компании Nvidia. Например, в Ubuntu 12.04 это делается командой

```
sudo aptitude install nvidia-experimental-310
```

При этом устанавливается драйвер версии 310.14. Здесь и далее мы ориентируемся именно на этот дистрибутив. В случае несовпадении предпочтений с нашими в сети легко найти пошаговую инструкцию фактически для любого другого дистрибутива.

После осознания того, что ваш домашний компьютер теперь годен для установки CUDA, нужно закачать двоичный установщик с сайта разработчика https://developer.nvidia.com/cuda-downloads. Для Ubuntu 12.04 годится бинарник, собранный для версии 11.10. Здесь же нужно выбрать между установочными файлами 64-bit и 32-bit. Далее будем действовать в предположении, что у вас 64-битная версия дистрибутива. Размер установочного файла порядка 670 МБ, но качается он достаточно бодро.

Чтобы иметь возможность скомпилировать поставляемое с дистрибутивом CUDA, нужно удостовериться в наличии следующих пакетов:

```
sudo aptitide install freeglut3-dev build-essential libx11-
dev  libxmu-dev libxi-dev libgl1-mesa-glx libglu1-mesa
libglu1-mesa-dev
```

графику

Если у вас 64-битная версия дистрибутива, то для установки примеров необходимо добавить символическую ссылку на библиотеку *libglut.so*.

```
sudo ln -s /usr/lib/x86_64-linux-gnu/libglut.so.3 /usr/lib/libglut.so
```

Установочный скрипт ищет эту библиотеку почему-то в **/usr/lib/**.

Теперь запускаем программу установки:

```
chmod +x cuda_5.0.35_linux_64_ubuntu11.10-1.run
sudo ./cuda_5.0.35_linux_64_ubuntu11.10-1.run
```

Аккуратно пролистываем и соглашаемся с EULA, отказываемся от установки драйвера для видеокарты, но соглашаемся с установкой CUDA и примеров. По умолчанию установка идет в **/usr/local/**, и если умолчание не менялось, то для начала работы достаточно переопределить переменные окружения:

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64
```

Перед началом экспериментов полезно побродить по директории **/usr/local/cuda**, посмотрев на 100 МБ документации в директории **doc** и поковырявшись в примерах из директории **samples**. Примеры собираются с помощью команды **make**. В директории **samples/0_Simple/template** располагается заготовка для стандартного CUDA-проекта.

Исходники и их компиляция

Программная часть архитектуры CUDA описывает расширение языка C/C++, функциональность и ключи управления работой компилятора *nvcc*, интерфейс оболочки CUDA Runtime системного драйвера графического ускорителя, профилировщик и отладчик.

Во время исполнения на центральном процессоре компьютера программа запускает функции на графическом ускорителе. Таких функций в программе может быть несколько. Функция, исполняемая на графическом ускорителе, называется ядром [kernel]. Во время запуска ядра порождается множество потоков, которые будут исполнять одну и ту же функцию на процессорах графического ускорителя. Поведение вычислительных потоков зависит как от кода функции, так и от их координат.

Файлы исходников имеют расширение **.cu**:

```
// example1.cu
#include <stdio.h>
#include <cuda_runtime_api.h>
int main() {
  int N;
  cudaDeviceProp prop;
  // Подсчитываем число устройств CUDA
  cudaGetDeviceCount(&N);
  for (int i = 0; i < N; i++){
    // Получаем информацию об устройстве
    cudaGetDeviceProperties(&prop, i);
    // Выводим иформацию об устройстве
    printf("Устройство N %d: %s\n",i+1,prop.name);
  }
}
```

Сборка производится с помощью компилятора *nvcc*:

```
nvcc example1.cu -o example1
./example1
Устройство N 1: GeForce GTS 450
```

В приведенном выше примере пока нет описания ядра, и все исполнение идет на центральном процессоре без привлечения ускорителя.

Расширение языка C/C++ позволяет описывать в одном исходном файле как основную программу, так и ядра с переменными и массивами, которые будут располагаться в памяти графического ускорителя. Кроме того, в расширении языка имеется компактная конструкция для запуска ядер, скрывающая вызов функции драйвера и упаковку аргументов.

Взаимодействие с графическим ускорителем осуществляется через интерфейс CUDA Runtime системного драйвера: запуск ядра, динамическое выделение памяти в графическом ускорителе, копирование данных из/в память графического ускорителя.

Компилятор *nvcc* разделяет входной файл на две части: одна будет исполняться на центральном процессоре и компилироваться стандартным компилятором gcc, а вторая – на графическом ускорителе, и компилироваться самим *nvcc*. Расширениями CUDA языка C/C++ можно пользоваться только в файлах с расширением **.cu**.

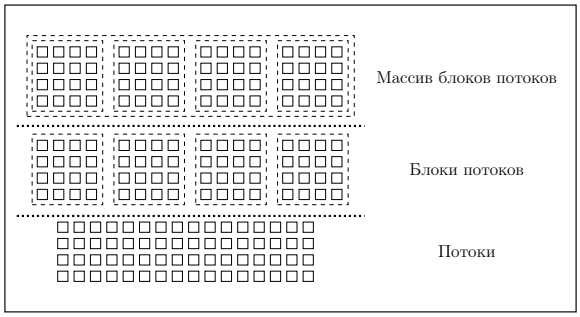
Иерархическая организация потоков

При программировании графического ускорителя необходимо быть в курсе диктуемой архитектурой иерархии исполняемых потоков и памяти. Это ключ к ускорению параллельных вычислений. К сожалению, в этом месте очень легко ошибиться и убить весь эффект большого числа процессоров неэффективным копированием данных.

При запуске программы на графическом ускорителе порождается множество потоков [thread]. Все потоки поделены на группы одного размера – блоки потоков [block]. Максимальный размер блока потоков на современных графических ускорителях равен 1024. У каждого потока и блока потоков имеются свои уникальные идентификаторы, называемые координатами. Тем самым, для разных потоков аргументы исполняемых инструкций и их последовательность могут различаться, поскольку могут зависеть от координат потока и блока потоков. Множества координат потоков и блоков потоков образуют одно-, дву- или трехмерные массивы-сетки [grid]. Размеры блока потоков и массива блоков потоков задаются при запуске ядра.

Во время исполнения ядра потоки одного блока могут синхронизироваться между собой посредством барьеров (механизм __syncthreads()), а потоки разных блоков исполняются независимо. Кроме возможности барьерной синхронизации, потоки одного блока могут взаимодействовать посредством разделяемой памяти. Потоки разных блоков могут взаимодействовать лишь через глобальную память, аналог оперативной памяти в компьютере. Кроме разделяемой и глобальной, есть константная и текстурная памяти, которые доступны из потоков только на чтение. На аппаратном уровне глобальная, константная, текстурная и разделяемая памяти оптимизированы под различные варианты использования. »

» **Не хотите пропустить номер?** Подпишитесь на www.linuxformat.ru/subscribe/!



› Иерархическая организация потоков.

Внутри каждого потока доступны структуры, которые позволяют его идентифицировать:

- **threadIdx** – координаты потока в блоке потоков;
- **blockIdx** – координаты блока потоков в сетке;
- **blockDim** – размеры блока потоков;
- **gridDim** – размеры сетки блоков потоков.

Азбука вызовов

В этом разделе перечислены простейшие языковые структуры, которые смогут пригодиться в процессе введения в среду CUDA.

Функции

Описание произвольной функции может предваряться следующими ключевыми словами:

- **__global__** – это функция-ядро, которая запускается на графическом ускорителе. Тип возвращаемого результата должен быть void.
- **__device__** – функция, которая вызывается из ядра.
- **__host__** – функция, которая вызывается с центрального процессора.

Возможно одновременное использование __device__ и __host__ – это означает, что функция может быть вызвана как из ядра, так и из программы на центральном процессоре. По умолчанию, при отсутствии вышеперечисленных ключевых слов, считается, что функция будет вызываться только с центрального процессора (__host__).

Функция-ядро декларируется примерно так:

```
__global__ void MyKernel(int *a,int *b,int *c) {
  int i = threadIdx.x + blockDim.x * blockIdx.x;
  c[i] = a[i] + b[i];
}
```

Здесь в качестве аргументов передаются три указателя в область памяти видеокарты. Каждый порожденный поток складывает i-е элементы массивов a и b и записывает результат в массив c; номер элемента в массиве i вычисляется в зависимости от координаты потока threadIdx.x, блока потоков blockIdx.x и размера блока потоков blockDim.x по оси Ox.

То же самое можно изобразить и с помощью вспомогательной функции __device__:

```
__device__ int MyDev(int a,int b) {
  return a + b;
}
__global__ void MyKernel(int *a,int *b,int *c) {
  int i = threadIdx.x + blockDim.x * blockIdx.x;
  c[i] = MyDev( a[i], b[i] );
}
```

Чтобы вызвать функцию-ядро, необходимо указать размер блока потоков и размер сетки блоков. Для этого используются тройные угловые скобки:

```
MyKernel<<< 256, N/256 >>>( a, b, c );
```

В случае порождения дву- или трехмерных сеток, что удобно при расчете двумерных и трехмерных задач соответственно, для передачи информации о размере размерах блока и сетки используется специальный тип данных – dim3:

```
__global__ void MyKernel2D(int *a, int *b, int *c) {
  int x = threadIdx.x + blockIdx.x * blockDim.x;
  int y = threadIdx.y + blockIdx.y * blockDim.y;
  int i = y * Nx + x; c[i] = b[i] + a[i];
}
dim3 blockSize( 16, 16, 1 );
dim3 gridSize( Nx/16, Ny/16, 1 );
MyKernel2D<<< blockSize, gridSize >>>( a, b, c );
```

Здесь происходит сложение двух матриц размера Nx×Ny. Количество порождаемых потоков в точности соответствует количеству элементов в матрице.

Работа с памятью

При объявлении переменной для ее размещения в глобальной памяти используется ключевое слово __device__, для размещения в разделяемой памяти – __shared__, а для размещения в константной – __constant__.

Объявленные внутри ядра переменные без использования этих трех ключевых слов автоматически отображаются на регистры или локальную память (область из глобальной памяти, область видимости – поток).

Текстуры объявляются с помощью шаблона texture< >. Размер и область глобальной памяти, к которой будет привязана текстура, определяются с помощью вызова cudaBindTexture().

Функция cudaGetSymbolAddress() через первый аргумент возвращает адрес переменной или массива, объявленного с помощью __device__ или __constant__:

```
texture< float, 1, cudaReadModeElementType > tex;
__device__ int dev_c;
__device__ float dev_a[ 1024*256 ];
__constant__ float dev_g[ 1024*256 ];
__global__ void MyKernel( float * c ){
  int i = threadIdx.x + blockDim.x * blockIdx.x;
  __shared__ int sb[ 256 ];
  sb[ threadIdx.x ] = dev_g[ i ] + dev_c;
  __syncthreads();
  c[i] = sb[threadIdx.x] + tex1Dfetch(tex, i) + dev_g[i];
}
int main(){
  float *p_dev_a;
  cudaGetSymbolAddress( &p_dev_a, dev_a );
  cudaBindTexture( 0, tex, p_dev_a, tex.channelDesc,
1024*256*sizeof(float) );
  MyKernel <<< 1024, 256 >>>( dev_c );
}
```

Функция cudaMalloc позволяет выделить область в глобальной памяти графического ускорителя указанного размера и возвращает указатель на эту область через первый аргумент:

```
cudaError_t cudaMalloc( void **, size_t );
```

При этом значение самого указателя будет храниться в оперативной памяти компьютера:

```
int main(){
  float *dev_a;
  cudaMalloc( &dev_a, sizeof(int)*1024*256 );
  cudaBindTexture( 0, tex, dev_a, tex.channelDesc,
1024*256*sizeof(float) );
  MyKernel <<< 1024, 256 >>>( dev_c );
  cudaFree( dev_a );
```

» **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

```
}
```

Функция cudaMallocPitch выделяет область памяти для расположения в ней двумерных массивов. При этом увеличивается размер строки в байтах до ближайшего числа, кратного 128, чтобы каждая строка двумерного массива начиналась со 128-байтного сегмента памяти. Это связано с оптимизацией доступа в память:

```
cudaError_t cudaMallocPitch( void** p_dev, size_t* pitch size_t
width, size_t height );
```

где p_dev – возвращаемый адрес выделенной памяти, pitch – возвращаемый новый размер строки в байтах, width – размер исходной строки в байтах, а height – количество строк.

Копирование памяти осуществляется с помощью семейства функций cudaMemcpy:

```
cudaError_t cudaMemcpy(void* dst, const void* src, size_t
count,enum cudaMemcpyKind kind);
cudaError_t cudaMemcpyToSymbol(const char* dst, const void*
src, size_t count, size_t offset=0);
cudaError_t cudaMemcpyFromSymbol(void* dst, const char* src,
size_t count, size_t offset=0);
cudaError_t cudaMemcpy2D(void* dst, size_t dpitch, const
void* src, size_t spitch, size_t width, size_t height, enum
cudaMemcpyKind kind);
```

где dst определяет область памяти, куда копируется сегмент памяти, определяемый переменной src. В функции cudaMemcpy сегменты памяти определяется через указатели. В функциях же cudaMemcpyToSymbol и cudaMemcpyFromSymbol одна из областей является переменной или массивом, определенным с помощью ключевых слов __device__ или __constant__. Функция cudaMemcpy2D копирует двумерный массив, память под который была выделена с помощью вызова cudaMallocPitch.

Через аргумент count указывается количество копируемых байт, а через offset – смещение относительно начала. Последний аргумент kind указывает направление копирования, возможные значения которого:

- cudaMemcpyDeviceToHost
- cudaMemcpyHostToDevice
- cudaMemcpyDeviceToDevice

где Device – память графического ускорителя, а Host – память компьютера.

Ожидание завершения

Запуск ядер осуществляется в асинхронном режиме. Это означает, что после постановки ядра драйвером в очередь на запуск управление тут же возвращается в программу. То есть функция cudaMalloc, вызванная сразу после вызова ядра, скопирует не те данные, которые ожидаются, поскольку ядро не завершило работу. Чтобы дождаться исполнения всех запущенных ядер, необходимо воспользоваться функцией cudaDeviceSynchronize:

```
int main(){
  float *dev_a, *dev_b;
  float *host_a;
  host_a = (float*)malloc(sizeof(float) * 1024*256);
  cudaMalloc( &dev_a, sizeof(float)*1024*256 );
  cudaMalloc( &dev_b, sizeof(float)*1024*256 );
  cudaMemcpy( dev_a, host_a, sizeof(float)*1024*256,
cudaMemcpyHostToDevice );
  MyKernel <<< 1024, 256 >>>( dev_a, dev_b );
  cudaDeviceSynchronize();
  cudaMemcpy( host_a, dev_a, sizeof(float)*1024*256,
cudaMemcpyDeviceToHost );
  cudaFree( dev_a );
  cudaFree( dev_b );
  free( host_a );
}
```

Проверка ошибок

Любая функция CUDA Runtime возвращает значение типа cudaError_t, сигнализирующее об успешности выполнения. Функция cudaGetLastError возвращает результат последней вызванной функции CUDA Runtime или ядра. Функция cudaGetErrorString возвращает строку, расшифровывающую ошибку.

Для удобства можно определить следующий макрос проверки ошибок:

```
#define cudaCheck {
  cudaError_t err = cudaGetLastError();
  if ( err != cudaSuccess ){
    printf(« cudaError = '%s' \n in '%s' %d\n»,\
cudaGetErrorString( err ),\ __FILE__, __LINE__ );
    exit(0);\
  }
}
```

Этот макрос проверяет, была ли ошибка, а если была, то выводит на печать описание, имя файла и номер строки, где возникла ошибка. За счет проверки каждого вызова функций CUDA Runtime и ядер удается найти большую часть ошибок на ранних стадиях разработки программ.

Время работы?

Измерение времени работы ядра с помощью системных функций не точно, поскольку оно включает время, затрачиваемое на взаимодействие программы с драйвером. Для точного измерения времени работы ядер определены специальный тип события cudaEvent_t и функции работы с ним:

```
cudaError_t cudaEventCreate (cudaEvent_t *event)
cudaError_t cudaEventRecord (cudaEvent_t event)
cudaError_t cudaEventSynchronize (cudaEvent_t event)
cudaError_t cudaEventElapsedTime (float *ms, cudaEvent_t start,
cudaEvent_t end)
```

Здесь cudaEventCreate инициализирует переменную типа cudaEvent_t, cudaEventRecord устанавливает событие, cudaEventSynchronize дожидается завершения события, а cudaEventElapsedTime измеряет время между двумя событиями в миллисекундах. Функция cudaEventElapsedTime выдает ошибку, если события start или end не завершились.

Пример программы с измерением времени работы ядра:

```
cudaEvent_t start, end;
float time;
cudaEventCreate( &start );
cudaEventCreate( &end );
cudaEventRecord( start );
MyKernel<<< 1024, 256 >>>();
cudaEventRecord( end );
cudaEventSynchronize( end );
cudaEventElapsedTime( &time, start, end );
printf(«Time: %.2f\n», time/1000.0 );
```

Заключение

Это лишь малая часть технической информации, которая потребуется на пути изучения CUDA. С другой стороны, пользовательский интерфейс – это не ракетные технологии, и его вполне можно освоить за разумный промежуток времени. LXF

Обратная связь

Приглашаем высказаться потенциальных авторов статей по параллельным вычислениям – ценные предложения, критику и советы присылайте по электронной почте: kalgin@ssd.sscc.ru, E.M.Baldin@inp.nsk.su.

VirtualBox: 6 XP

Максим Черепанов излагает, как построить и администрировать офисные рабочие места с минимумом затрат и максимумом эффективности.

Наш эксперт

Максим Черепанов Замдиректора небольшой коммерческой фирмы. Перешел в Linux в 2008 г. и уже успел разучиться чистить реестр и забыть слово «антивирус».

Случилось так, что передо мной встала следующая задача: организовать переезд «полевого», или удаленного (т. е. не подключенного к основной локальной сети), филиала некой фирмы, из 8 рабочих станций. Все они были настроены и работали на разных версиях разных ОС на разномастном реальном железе, которое глючило по-разному. 6 рабочих мест были типовые офисные, т. е. на них накладывались стандартные требования: почта, офис, совместное пользование файлами, печать документов; причем на трех из них должен был быть доступ в Интернет. Остальные два работали с Windows XP на реальном железе: на одном – несколько лицензионных программ и термопринтер этикеток, на втором 1С.

Но все это было в старом помещении, а при переезде в новое неразумно тащить за собой старые проблемы. Заручившись согласием руководства на некоторые дополнительные траты, за 2 месяца до переезда я стал готовить железо и софт.

Оборудование и сетевая инфраструктура

Типовые рабочие места я сразу же решил сделать виртуальными, создав их на сервере, к которому они должны будут подключаться с терминалов. К началу проекта были в наличии:

- Сервер для доступа во внешний мир. Железо – Intel® Core™ 2 Duo, 2 Ethernet, на одном ppp на оптическом модеме, второй со статическим IP смотрит внутрь локальной сети. Софт – совершенно стандартный набор: Ubuntu 10.04 Server, SSH, *Webmin*, *Squid*, *Shorewall*, *Postfix*. Этот компьютер успешно работал в прежнем помещении, система была переустановлена заново. Внутренний статический IP – 192.168.0.200.
- Два обычных компьютера (перенесены из старого помещения). ОС на них возвращена к исходному виду из слепков, все программы переустановлены заново.
- Несколько принтеров – сетевых и с общим доступом.
- Шесть лицензий Windows XP PRO SP3 Get Genuine Kit. Строго говоря, в них не было необходимости, но они были уже куплены до реконструкции филиала. Не солить же их!

Для доукомплектации потребовались следующие приобретения:

- Для 6 типовых рабочих мест – терминалы Neoware CA-21: VIA Eden 800 МГц CPU, 512 МБ ОЗУ, 1 ГБ Flash, 4 порта USB, VGA, LAN. Короче, все что надо для нормальной работы. По случаю получилось купить на аукционе eBay – в пересчете на российские рубли, по 700 руб./шт.
- Новые клавиатуры и мыши с разъемами PS для полной совместимости с терминалами. Хотя потом выяснилось, что они также прекрасно работают с USB-периферией. Приобрел скорее для того, чтобы все было чисто и красиво – прежняя периферия работала, но была изношена практически до предела.
- Компьютер для сервера виртуальных машин Intel® Core™ i3-2100 CPU @ 3.10 ГГц, 4 ядра, 12 ГБ оперативной памяти, 1-ТБ винчестер. Видео и сеть интегрированные. Софт: Ubuntu-12.04 Server, в ней запущен *VirtualBox* с дополнительными пакетами, внутри которого крутятся 6 виртуальных ядер Windows XP (рабочие станции), *Samba* для общего пользования файлами, система запуска ядер, система загрузки терминалов промежуточной ОС. Внутренний IP – 192.168.0.250.

Настройка сервера виртуальных машин

Исходное состояние – свежеустановленный Ubuntu 12.04 Server; конечно, установлены и настроены служба SSH и файловый менеджер *mc*.

На сайте VirtualBox заходим в https://www.virtualbox.org/wiki/Linux_Downloads. Выбираем свой дистрибутив и добавляем в **/etc/apt/sources.list** нужную строку:

```
deb http://download.virtualbox.org/virtualbox/debian precise contrib
```

Скачиваем, сохраняем и устанавливаем публичный ключ Oracle:

```
wget -q http://download.virtualbox.org/virtualbox/debian/oracle_vbox.asc -O- | sudo apt-key add -
```

Обновляем репозитории и устанавливаем *dkms* и *virtualbox*:

```
sudo apt-get update
sudo apt-get install dkms virtualbox-4.2
```

Переходим на https://www.virtualbox.org/wiki/Downloads, копируем ссылку на текущую версию Oracle VM Virtualbox Extention Pack и закачиваем его:

```
wget http://download.virtualbox.org/virtualbox/4.2.6/Oracle_VM_VirtualBox_Extension_Pack-4.2.6-82870.vbox-extpack
```

в одном Ubuntu

Этот пакет позволяет использовать RDP и USB 2.0.

```
$ sudo vboxmanage extpack install ~/Oracle_VM_VirtualBox_
Extension_Pack-4.2.6-82870.vbox-extpack
```

Во время установки *VirtualBox* добавится новая группа vboxusers. Пользователя, от имени которого будет запускаться *VirtualBox*, нужно добавить в эту группу, иначе запуск будет невозможен:

```
$ sudo usermod -aG vboxusers ian
```

Здесь мы добавили в группу vboxusers пользователя ian.

2 Устанавливаем *unzip*, *Apache* и пакет для поддержки PHP:

```
$ sudo apt-get install unzip apache2 libapache2-mod-php5
```

После установки *Apache* должен запуститься. Если этого не произошло, на всякий случай перезапустите его:

```
$ sudo /etc/init.d/apache2 restart
```

3 Переходим на http://code.google.com/p/phpvirtualbox/, скачиваем и распаковываем *phpvirtualbox* необходимой версии:

```
# wget http://phpvirtualbox.googlecode.com/files/
phpvirtualbox-4.2-4.zip
```

В архиве имеется папка **phpvirtualbox-*.*** – эту папку (не ее содержимое, а именно ее) переносим в **/var/www** и переименовываем в **/var/www/phpvirtualbox**. Переименовываем внутри этой папки файл **config.php-example** в **config.php** и редактируем его – надо указать логин пользователя с правами администратора, от имени которого будет запускаться *VirtualBox*, а также его пароль:

```
var $username = 'ian'
var $password = 'pass'
```

и выбрать русский язык для интерфейса:

```
var $language = 'ru'
```

4 Создаем конфигурационный файл для запуска *vboxweb-service*:

```
$ sudo touch /etc/default/virtualbox
```

Его минимальное содержание –

```
VBOXWEB_USER=ian
```

а именно логин пользователя, имеющего административные права, от имени которого будет запускаться VirtualBox.

Описание других опций можо посмотреть на http://code.google.com/p/phpvirtualbox/wiki/vboxwebServiceConfigLinux.

5 Стартуем сервис:

```
$ sudo /etc/init.d/vboxweb-service start
```

Открываем браузер на своем компьютере и пишем в адресной строке: http://имя_сервера/phpvirtualbox. Логин и пароль по умолчанию – admin и опять же admin.

На рис. 1 изображено то, что вы должны увидеть: такой же интерфейс, как у десктопной версии.

Устанавливаем столько и такие виртуальные машины, как требуется. В моем случае это шесть Windows XP. Я сделал одну машину, а все остальные клонировал с нее. Обязательно устанавливаем дополнения гостевой системы.

6 Организуем автозапуск некоторых виртуальных машин одновременно с host-сервером и автоматическое корректное выключение виртуалок вместе с сервером.

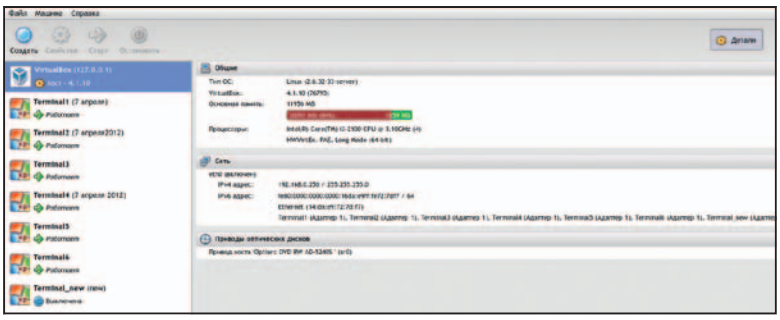


Рис. 1. Вот как выглядит PHP-интерфейс *VirtualBox*.

Забираем скрипт на странице: http://mmm-dev.blogspot.ru/2011/04/virtualbox.html. Сохраняем его в **/etc/init.d** под именем **vboxcontrol** и делаем исполняемым:

```
$ sudo chmod +x /etc/init.d/vboxcontrol
```

Создаем конфигурационные файлы **/etc/virtualbox/config** и **/etc/virtualbox/machines_enabled**.

В первом указывается пользователь, от имени которого запускается скрипт (с административными привилегиями):

```
VM_USER=ian
```

Во втором перечисляются виртуальные машины, которые будут автоматически стартовать вместе с хостом (по имени машины, каждая с новой строки):

```
XP_main
XP_1
XP_2
```

Осталось обновить скрипты **init.rc** –

```
$ sudo update-rc.d vboxcontrol defaults 99 10
```

после чего становятся доступны команды **service vboxcontrol start|stop|status|savestate|poweroff** (для виртуальной машины из списка), а также **service vboxcontrol vm-start|vm-stop|vm-poweroff|vm-savestate|vm-reset VM_NAME** (для конкретной виртуальной машины).

Настройка тонких клиентов

По большому счету надо только включить настройку по сети в BIOS (будьте внимательны, обычно она включается в двух местах BIOS'а. Я для удобства записываю на внутреннюю флэш-память еще какой-нибудь маленький Linux типа Puppy или CorePlus) и, конечно, записать MAC-адрес Ethernet карты терминала. Допустим, он 00:60:08:C7:A3:D8 – обычно он записан на корпусе, или его можно узнать, набрав в консоли терминала **ifconfig** (вот еще для чего я ставлю внутрь маленький Linux: мало ли что написано на корпусе).

Теперь настроим загрузки по сети в сервере виртуальных машин.

1 Установим на сервер виртуальных машин демон TFTP:

```
$ sudo apt-get install tftp tftpd
```

По умолчанию TFTP-сервер настроен на использование директории **/srv/tftp**. Мы перестроим его на использование **/tftpboot**: в этой директории будут храниться файлы, которые мы можем скачать с сервера или закачать в него.

» **Не хотите пропустить номер?** Подпишитесь на www.linuxformat.ru/subscribe/!

Отредактируем файл **/etc/inetd.conf**, найдя строки такого вида:

```
#:BOOT: TFTP service is provided primarily for booting. Most sites
#       run this only on machines acting as "boot servers".
tftp        dgram  udp    wait    nobody  /usr/sbin/tcpd  /usr/sbin/in.tftpd /srv/tftp
```

Последняя строка должна получиться такой:

```
tftp        dgram  udp    wait    nobody  /usr/sbin/tcpd  /usr/sbin/in.tftpd /tftpboot
```

Создадим рабочую директорию и поменяем права доступа:

```
$ sudo mkdir /tftpboot
$ sudo chown -R nobody /tftpboot
```

Перезапустим сервер *inetd*:

```
$ sudo /etc/init.d/openbsd-inetd restart
```

Поместим в директорию **/tftpboot** какой-нибудь файл (**myfile**) и выполним команду

```
$ tftp localhost
```

На приглашение TFTP-сервера введем

```
tftp> get myfile
```

Если файл получен, значит, все правильно, и перед вами появится сообщение

```
Received 168 bytes in 0.1 seconds
```

Если вы не видите этого сообщения, проверяйте все заново.

2 Установим сервер DHCP и настроим его для удаленной загрузки терминалов.

```
$ sudo apt-get install dhcp3-server
```

Отредактируем **/etc/dhcp3/dhcpd.conf**:

```
ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;
authoritative;
log-facility local7;
server-name “192.168.0.200”; # Адрес DNS сервера,
у меня это другая машина
subnet 192.168.0.0 netmask 255.255.255.0 {
pool {
range 192.168.0.170 192.168.0.190; # Диапазон работы DHCP,
остальные машины у меня со статическими IP
next-server 192.168.0.250; # Адресе tftp сервера с загрузчиком
filename “pxelinux.0”; # Имя файла с загрузчиком
}
}
```

Перезапустим DHCP-сервер:

```
$ sudo /etc/init.d/dhcp3-server restart
```

3 Наполняем директорию **/tftpboot**: заходим на http://www.kernel.org/pub/linux/utils/boot/syslinux/, забираем и распаковываем **syslinux**, оттуда будем брать необходимые файлы.

Создадим директорию **/tftpboot/boot** и загрузим в нее файл **vesamenu.c32** (для создания и редактирования загрузочного меню).

В директорию **/tftpboot** загрузим файл **pxelinux.0** (сетевой загрузчик).

Создадим директорию **/tftpboot/pxelinux.cfg**, а в ней – файл **default** следующего содержания:

```
default /tftpboot/boot/vesamenu.c32
menu color border 0 #ffffffff #00000000
menu color sel 7 #ffffffff #ff000000
menu color title 0 #ffffffff #00000000
menu color unsel 0 #ffffffff #00000000
menu color disabled 0 #ffffffff #00000000
menu color timeout_msg 0 #ffffffff #00000000
menu color timeout 0 #ffffffff #00000000
menu background #1F45A5
menu title PXE Network Boot Menu
prompt 0
timeout 20
label Thinstation
  kernel /tftpboot/pxe/thinstation/vmlinuz
  append ramdisk_blocksize 4096 initrd=/tftpboot/pxe/thinstation/initrd root=/dev/ram0 ramdisk_size=524288 splash=silent vga=791 console=ttyS3
label Boot from first hard disk
        localboot 0x80
menu end
```

Заходим на http://www.thinstation.org/, забираем и распаковываем дистрибутив Thinstation. Сейчас уже доступна версия 5, но я использую 2.5, поэтому мне неизвестно, будет ли работать все перечисленное на новой версии.

Создадим директорию **/tftpboot/pxe/thinstation**, загрузим в нее файлы **initrd** и **vmlinuz** из выбранного дистрибутива Thinstation.

В директории **/tftpboot** создадим файл **thinstation.hosts** соответствия имен и MAC-адресов для всех тонких клиентов, следующего содержания:

#Имя_компьютера	Mac_адрес	Имя_группы	Коментарии
terminal1	006008C7A3D8	Firma	#
terminal2	000000000000	Firma	#
terminal3	000000000000	Firma	#
terminal4	000000000000	Firma	#
terminal5	000000000000	Firma	#
terminal6	000000000000	Firma	#

Первый MAC-адрес – из нашего примера; видоизмените файл по вашим данным.

Для каждого выбранного вами имени тонкого клиента (первый столбец) создайте в директории **/tftpboot** собственный конфигурационный файл (**thinstation.conf-terminal1** для имени terminal1 и подобные файлы для других терминалов):

```
AUTOSTART=On
SCREEN_RESOLUTION="1024x768"
SCREEN_COLOR_DEPTH="16"
RDESKTOP_COMPRESSION=On
RDESKTOP_COLOR_DEPTH="16"
KEYBOARD_MAP=en_us # Раскладка клавиатуры
TIME_ZONE="Europe/Moscow"
USB_ENABLED=On # Включение драйверов USB, нужно для клавиатуры и мыши
AUDIO_LEVEL=67
SESSION_0_TITLE="Terminal 1"
SESSION_0_TYPE=rdesktop
SESSION_0_RDESKTOP_SERVER=192.168.0.250:3391 # RDP-порт виртуальной машины
SESSION_0_RDESKTOP_OPTIONS=”-u user -z”# user – под этим логином будет предпринята попытка входа в сессию RDP виртуальной машины
```

Обратите внимание на предпоследнюю строку: данный порт надо будет прописать в свойствах нужной вам виртуальной машины, позже я напомню про это.

Если все сделано верно, то можно пробовать запускать терминалы. Первым пойдет загрузка PXE, определится сервер DHCP, терминал получит IP, затем пойдет загрузка из **/tftpboot**, на 2 секунды выскочит загрузочное меню и начнется загрузка Thinstation. Поскольку RDP в виртуальных машинах еще не включен, загрузка закончится серым экраном с двумя опциями: Shutdown и Reconnect.

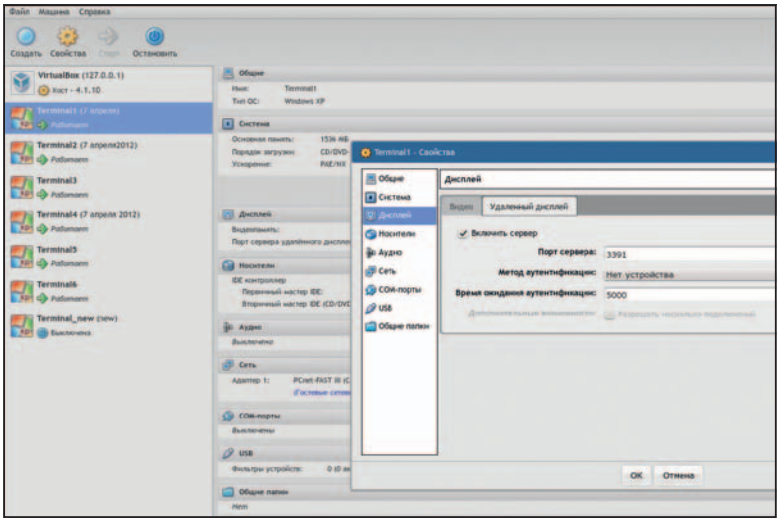
» **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

Настройка виртуальных машин

Будем считать, что все сказанное выше прошло успешно, и виртуальные машины в потребном количестве успешно инсталлированы через *phpvirtualbox*. Про настройку памяти, сети и прочего виртуального оборудования писать не буду, этого вполне достаточно в Сети. Остановлюсь на активации RDP. Для него обязателен пакет расширения *Oracle VM Virtualbox Extention Pack*.

На рис. 2 показано, как правильно сделать настройку RDP: поставить галочку и прописать тот порт, который указан в конфигурации. Обратите внимание, что IP при обращении по RDP у виртуальных машин тот же, что и у сервера виртуальных машин, но порт у каждой свой. Это важно помнить, если надо зайти удаленно не с терминала, который автоматически подгружает конфигурацию, а вручную с другого компьютера для помощи пользователю.

Все, перезагружаем север виртуальных машин. Проверяем (заходом на http://имя_сервера/phpvirtualbox), стартовали ли машины из списка. Если да – перезагружаем тонкие клиенты, для которых настроены конфигурационные файлы. У меня через 65 секунд после включения питания появляется окно приглашения входа в Windows.



› Рис. 2. Настройка RDP к виртуальным машинам.

Дополнения и замечания

1 Доступ к *phpvirtualbox* организован средствами *Shorewall*, в **rules** добавлена следующая строка:

```
Web/DNAT      net    loc:192.168.0.250
```

Таким образом, обращаясь к порту 80 на внешний IP, попадаем на *Apache* внутреннего сервера виртуальных машин.

2 На сервере виртуальных машин запущена *Samba* для совместного пользования документами. Поскольку физически операционная система и документы находятся на одной машине, открытие документов происходит в разы быстрее, чем с физической машины, подключенной к этому же серверу по локальной сети.

3 Доступ в Интернет регулируется совершенно так же, как для физических компьютеров: с помощью *Squid* и статических адресов, которые прописаны в настройках самих виртуальных операционных систем.

4 Я не стал заморачиваться с пробросом портов принтеров и сканеров на терминалы. К двум имеющимся физическим машинам подключил потребное количество периферии проводами подлиннее, открыл к ним общий доступ, и все.

5 В *crontab* сервера виртуальных машин внесена следующая запись:

```
30 23 * * * /etc/init.d/vboxcontrol savestate
00 06 * * * /etc/init.d/vboxcontrol start
```

В 23:30 ежедневно виртуальные машины засыпают, в 6 утра просыпаются. В таком состоянии их можно клонировать или создать их резервные копии. Но поскольку у меня в заготовке есть настроенная полностью исправная машина, и время ее развертывания соизмеримо с развертыванием из ежедневной резервной копии, я не стал дополнительно нагружать *crontab*.

6 Если вы хотите удаленно помочь пользователю, можно поставить галочку «Разрешать несколько подключений» на свойствах дисплея в виртуальной машине (см. рис. 2, но на рисунке галочка не стоит). Я пробрасываю порт в *Shorewall* внешнего сервера – в файле **rules** вписываю строку

```
DNAT      net    loc:192.168.0.250:3391  tcp  11011
#RDESKTOP
```

и обращением по внешнему IP сервера на порт 11011 попадаю на рабочий стол терминала одновременно с пользователем.

7 Виртуальные машины можно тасовать, как карты в колоде, оперируя портами в RDP. Также, изменяя **/tftpboot/pxelinux.cfg/default**, можно создать большое меню для загрузки других ядер виртуальных машин. Информацию про это легко найти в Сети.

8 В качестве терминала можно использовать абсолютно любой компьютер с функцией загрузки по сети по спецификации PXE, нужно только вписать в общий конфиг MAC-адрес сетевой карты, жесткий диск на таком компьютере не обязателен.

9 Если вам попадется монитор с соотношением сторон 16:9, для нормального отображения придется немного изменить конфигурационный файл терминала:

```
AUTOSTART=On
X_MONITOR_MODELINE='"1366x768" 85.034880 1366 1424 1568
1776 768 772 776 798 -HSync +VSync'
SCREEN_RESOLUTION="1366x768"
SCREEN_HORIZSYNC="31.5-81"
SCREEN_VERTREFRESH="55-76"
I915RESOLUTION_ENABLED=On
I915RESOLUTION_OPTIONS="3c 1366 768 8"
#SCREEN_RESOLUTION="1024x768"
SCREEN_COLOR_DEPTH="16"
RDESKTOP_COMPRESSION=On
RDESKTOP_COLOR_DEPTH="16"
KEYBOARD_MAP=en_us # Раскладка клавиатуры
TIME_ZONE="Europe/Moscow"
USB_ENABLED=On # Включение драйверов USB
AUDIO_LEVEL=67
SESSION_0_TITLE="Terminal 1"
SESSION_0_TYPE=rdesktop
SESSION_0_RDESKTOP_SERVER=192.168.0.250:3391
SESSION_0_RDESKTOP_OPTIONS="-u user -z"
```

Кроме того, на сервере виртуальных машин надо явно задать разрешение для конкретной машины:

```
$ VBoxManage controlvm Terminal1 setvideomodehint 1366 768 16
```

Если у вас другие разрешение экрана и глубина цвета, поменяйте на свои (1366, 768 и 16) и после этого задайте нужные параметры экрана из виртуальной машины. Кажется достаточно сложным, но к сожалению, у меня без всех этих манипуляций нормального изображения на широкоэкранных мониторах не получалось.

Авторство

Из всего приведенного я не написал ни строчки, правку конфигурационных файлов под свои нужды написанием не считаю. Этот материал – просто сведение вместе известной информации. LXF

ЧАСТЬ 1

Crosstool-ng: Ищем

Дмитрий Кузнецов готовится к сборке собственного дистрибутива, тщательно подбирая инструменты.

Наш эксперт

Дмитрий Кузнецов Более 10 лет ведет неравный бой с монстрами сложности, хотя некоторые почему-то их считают ветряными мельницами.

Дистрибутив Linux ныне установить легко. За полчаса, ответив на ряд простых вопросов, любой новичок получает готовую ОС. Она сама загружается, снабжена графическим интерфейсом пользователя и содержит солидный набор прикладных программ, расширяемый несколькими движениями мыши. Это здорово! Разработчики дистрибутивов заслуживают самой искренней благодарности и восхищения. Большинству пользователей ничего сверх того и не надо. Но есть и те, кто хочет понимать внутреннее устройство системы, а не только пожинать плоды работы «черного ящика». Их пытливый ум вечно ставит вопросы. Как работает система? Из каких компонентов состоит? В каком порядке они запускаются при загрузке? Какие из них главные, а какие второстепенные? Как их собрать из исходных кодов? Анатомия Linux в общих чертах освещена в главной статье номера, но тут и целой книги будет мало.

Жертвовать глубиной изложения, говоря о таких фундаментальных проблемах, было бы ошибкой. Разумнее ограничить ширину охвата наиболее важными вопросами – их глубокое понимание станет прочной основой для дальнейшего изучения. Мы попробуем построить минимальный дистрибутив для архитектуры x86_64 и запустить его на виртуальной машине *KVM*. Эти базовые знания позволят читателям ладить с любым дистрибутивом Linux, а особо неугомонным – продолжить изыскания самим.

Комплектация

Очевидно, минимальный дистрибутив Linux должен уметь загрузить компьютер и предоставить пользователю возможность запускать имеющиеся в его составе программы. Какое ПО для этого нужно? Для ответа сперва разберемся в процессе загрузки Linux.

После включения питания процессору подается сигнал RESET, чтобы процессор установил свои регистры на передачу управления коду BIOS, записанному в ПЗУ. BIOS тестирует и инициализирует аппаратные устройства, а затем просматривает первые (загрузочные) секторы всех дисков, указанных в BIOS Setup, для поиска загрузчика ОС. Когда загрузчик найдется, он скопируется в оперативную память и начинает выполняться. Его задачи – найти ядро Linux, скопировать его в память и передать туда управление. Ядро выполняет свою инициализацию, монтирует корневую файловую систему и запускает первую пользовательскую программу *init* – обычно это **/sbin/init**, но в параметрах ядра можно указать любую другую. Действия *init* настраиваются в файле **/etc/inittab** – обычно указывается запуск стартового сценария и командной оболочки [shell]. Все, ОС загружена и ждет команд пользователя.

Теперь очевиден список необходимого ПО:

- Загрузчик и ядро Linux.
- *Init*, *shell* и другие прикладные программы.
- Библиотеки, необходимые для прикладных программ. Для большинства утилит командной строки Linux требуется только стандартная библиотека языка C (*GNU libc* или ее аналоги).
- Файл **/etc/inittab** для настройки *init*.
- Стартовый сценарий (запускается из *inittab*).

Для других архитектур (не x86) процесс загрузки отличается лишь начальной стадией: до передачи управления ядру Linux. Например, на встраиваемых платформах (архитектура ARM) часто используют *U-boot*. Это специальная стартовая среда (еще ее называют bootstrap-среда или загрузчик), которая выполняет функции BIOS и загрузчика на ПК.

Инструменты сборки

Комплектация дистрибутива определена. Исходные тексты компонентов доступны. Как их собрать? Ответ вроде очевиден: есть же *GCC*, стандартный компилятор для ОС Linux. Верно. Но не все знают, что *GCC* – только вершина айсберга. Для компиляции и генерации исполняемого кода из исходных текстов его мало, нужен целый инструментарий [*англ.* toolchain – цепочка инструментов]. Он состоит из следующих компонентов:

- *GCC* (GNU Compiler Collection) – набор компиляторов (в том числе C и C++).
- *Binutils* – ассемблер, компоновщик и набор программ для обращения с объектным кодом.
- *GNU libc* (или ее аналог) – стандартная библиотека языка C.
- Заголовочные файлы ядра Linux.

Все эти инструменты входят в состав любого дистрибутива Linux в готовом к использованию виде. Но учтите, что они собраны с целью генерировать исполняемый код только для данного дистрибутива и данной архитектуры. То есть с их помощью можно собрать только клон дистрибутива своего ПК. И двоичная совместимость будет превосходная. Кстати, в разности инструментариев и кроются корни проблем двоичной несовместимости разных дистрибутивов Linux. А фаза Луны, как думают иные, тут ни при чем.

Собирать только клоны неинтересно, нужно докопаться до первоисточников. Похоже, вопросов становится все больше. Если какие-то компоненты инструментария не устраивают? Если нужно собрать инструментарий для другой архитектуры (не x86)? Если на ПК нужно создать среду сборки ПО для своего смартфона с ARM процессором? Все это возможно. Компоненты инструментария поддерживают массу разных архитектур и имеют огромное количество параметров сборки. Но нужно уметь их самостоятельно собирать из исходных кодов. Конечно, это непростая задача, которая требует времени, опыта и специальных знаний.

Технические условия

Данная статья содержит не только теоретические сведения, но и описание практических шагов, которые, вероятно, многие захотят повторить, поэтому разумно с самого начала определить некоторые параметры рабочей среды:

- Все работы ведутся на ПК с процессором Intel(R) Core(TM)2 Quad Q9550, 4 ГБ оперативной памяти, ОС Linux Ubuntu 12.04 LTS 64-bit. Для работы *KVM* важно, чтобы процессор имел аппаратную поддержку виртуализации.
- Необходимо соединение с Интернет для загрузки архивов с исходным кодом.
- Перед началом работы надо получить права root. В Ubuntu для этого используется команда

```
sudo su
```

- Для удобства и краткости изложения определим рабочую структуру каталогов, где будут храниться все результаты работы: каталог **/root/work** с подкаталогами **download** (загруженные из Интернет файлы), **src** (исходные коды собираемого ПО) и **files** (все остальное). Создать ее можно так:

```
mkdir /root/work
cd /root/work
mkdir download src files
```

инструментарий

К счастью, есть замечательный проект – *crosstool-ng*. С его помощью легко задать нужные параметры инструментария, загрузить из Интернета исходные коды его компонентов и автоматически их собрать. Кроме того, *crosstool-ng* ведет подробный журнал выполненных команд, анализируя который, можно детально разобраться в процессе сборки.

Crosstool-ng: установка

Загрузим архив исходных кодов *crosstool-ng* с сайта проекта:

```
wget -nd -P /root/work/download crosstool-ng.org/download/crosstool-ng/crosstool-ng-1.16.0.tar.bz2
```

Он имеет формат **tar.bz2**, традиционный для мира Linux, поэтому легко распаковывается в рабочий каталог:

```
tar -C /root/work/src -xjvf /root/work/download/crosstool-ng-1.16.0.tar.bz2
```

Для работы *crosstool-ng* требуется дополнительное ПО – зависимости. Установить его (если это не сделано ранее) можно так:

```
apt-get install bison flex gperf texinfo gawk libtool automake libncurses5-dev g++ subversion
```

Сборка проекта тоже не отличается особой оригинальностью, за вычетом ключа **enable-local**, которым нам сейчас удобно воспользоваться. Он информирует *crosstool-ng*, что пользователь не намерен устанавливать его в систему, а будет запускать локально из каталога с его исходными кодами. Таким образом, для удаления *crosstool-ng* достаточно будет просто удалить этот каталог.

```
cd /root/work/src/crosstool-ng-1.16.0/
./configure --enable-local
make
```

Сборка инструментария

Итак, *crosstool-ng* собран и готов к работе. Можно собирать инструментарий. Сборка инструментария состоит из двух этапов:

1. Подготовка файла **.config**. Это – конфигурация собираемого инструментария.
2. Запуск сборки по созданной конфигурации.

Файл **.config** похож на тот, что используется ядром Linux. Есть и аналогичная интерактивная система конфигурирования, запустив которую, можно создать конфигурацию с нуля или откорректировать уже существующую. По умолчанию считается, что **.config** находится в текущем каталоге. К счастью, в состав *crosstool-ng* (подкаталог **samples**) входит множество примеров конфигураций, которые служат хорошей основой для построения своих собственных. Они настолько разнообразны, что часто достаточно изменить всего лишь несколько параметров.

Промежуточные результаты своей работы (исходный код компонентов, объектные файлы, журнал сборки и т. д.) *crosstool-ng* хранит в текущем каталоге. Поэтому для сборки каждого инструментария удобно создавать отдельный каталог, делать его текущим и все команды запускать из него:

```
mkdir -p /root/work/files/toolchain/build
cd /root/work/files/toolchain/build
```

Среди доступных примеров легко найти подходящий для данного случая: **/work/src/crosstool-ng-1.16.0/samples/x86_64-unknown-linux-gnu/crosstool.config**. Чтобы им воспользоваться, нужно скопировать его в рабочий каталог под именем **.config** и запустить интерактивную систему конфигурирования:

```
cp ../../../src/crosstool-ng-1.16.0/samples/x86_64-unknown-linux-gnu/crosstool.config ./.config
/root/work/src/crosstool-ng-1.16.0/ct-ng menuconfig
```

Она показывает параметры инструментария в иерархическом виде. Многие из них снабжены справочной информацией (кнопка Help внизу экрана). Это удобно для изучения: так быстрее разобраться, что в данной конфигурации надо изменить.

» **Раздел Paths and misc options** [Пути и общие настройки всей системы]

Установить Try features marking EXPERIMENTAL. Это даст возможность выбрать новейшие версии компонентов инструментария.

Указать в Local tarballs directory путь **/root/work/download**. Там будут сохранены загруженные архивы с исходными кодами. При повторной сборке они не будут загружаться снова.

Указать в Prefix directory путь **/root/work/files/toolchain/result**. Там будет сохранен собранный инструментарий.

» **Раздел Target options** [Параметры целевой архитектуры]

Здесь все правильно. В Target architecture установлено значение x86, а в Bitness – 64-bit, как раз наш случай. Но богатство выбора архитектур впечатляет: девять штук! Например, можно выбрать arm и получить инструментарий для сборки ПО под смартфоны.

» **Раздел версий компонентов**

Что-то они староваты. Пожалуй, стоит выбрать поновее. (Например, можно ориентироваться на Ubuntu 12.04 LTS, где полный список установленного ПО можно получить командой **dpkg -l**.)

Пройдя по подпунктам меню, выберем версию ядра Linux: Operating system > Linux kernel version – укажем 3.2.25.

Теперь аналогичным образом выберем версию binutils: Binary utilities > binutils version – нас интересует 2.22 (EXPERIMENTAL).

Нам понадобится определенная версия *GCC*: C compiler > gcc version – выберем 4.6.3. Заодно в том же разделе C compiler нужно отключить поддержку языков Fortran и Java (раз уж мы решили собирать минимальный дистрибутив, обойдемся без излишеств).

Далее озаботимся выбором библиотеки языка C. Отметим, что существует несколько аналогов этой библиотеки. Конечно, *GNU C library* (*glibc*) – это классика. Но не все ею довольны, и таких с каждым годом все больше и больше: есть у старушки ряд проблем. Поэтому разработчики некоторых дистрибутивов решили сменить *glibc* на один из ее аналогов. Скажем, в Ubuntu применяют *Embedded GLIBC* (*eglibc*). Итак, идем по подпунктам: C-library > C library – выберем eglibc, C-library > eglibc version – выберем 2.15.

Версии сопутствующих библиотек (математические библиотеки, используемые в *GCC*) указываются в разделе Companion libraries: GMP version – 5.0.2, MPFR version – 3.1.0, PPL version – 0.11.2, CLog/ppl version – 0.15.11, MPC version – 0.9.

» **Раздел Debug facilities** Для простоты лучше отключить тут все.

Осталось закрыть систему конфигурирования (дважды нажав Esc), согласившись с сохранением изменений в файле **.config**. Конфигурация инструментария готова. Можно запускать сборку:

```
/root/work/src/crosstool-ng-1.16.0/ct-ng build
```

На это потребуется несколько десятков минут. *crosstool-ng* самостоятельно загрузит исходный код необходимых компонентов, соберет их и предоставит подробный отчет о проделанной работе (**build.log** в текущем каталоге).

Ура, наш инструментарий подготовлен! Теперь можно собирать компоненты дистрибутива из исходных кодов – чем мы и займемся 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. Назначение KDE средой по умолчанию в Mint
2. Настройку Unity
3. Парольную защиту Grub
4. Очистку дисковых кэшей
5. Имена хостов в Android
6. Автоматическое управление кулером
7. Проблемы резервного копирования

1 Мне, пожалуйста, KDE

У меня Linux Mint KDE 14, но в результате установки пакетов появилось окно входа в систему Gnome, а затем Gnome стала средой по умолчанию. Теперь при каждой загрузке системы приходится выбирать KDE из выпадающего списка. Выбранный вариант можно запомнить, но не более того. У меня еще есть Mint KDE 13, и он ведет себя точно так же. Как сделать KDE средой по умолчанию?

Джон Терразе [John Terrase]

Судя по всему, во время установки пакетов Gnome Вы переключили менеджер дисплея. Заглянув в файл **/var/log/apt/history.log**, Вы, скорее всего, обнаружите, что во время установки был установлен *GDM* или *LightDM*, и он выбрал сам себя в качестве менеджера по умолчанию.

Решение на удивление простое – удалять другой менеджер дисплея не нужно, так как при следующем обновлении он снова появится. Нужно лишь запустить команду

```
sudo dpkg-reconfigure kdm
```

Она выполнит повторную настройку пакета *kdm*, при которой менеджер *KDM* еще раз будет выбран менеджером по умолчанию.

При следующем входе в систему средой по умолчанию должна снова стать KDE, так как настройки *KDM* должны были сохраниться.

2 Пустая линза

Я обновил Ubuntu с 11.10 на 12.04, и линза приложений оказалась пустой, хотя линза файлов по-прежнему работала. Я переустановил систему с компакт-диска, но проблема сохранилась, и мне кажется, что она сидит в файле настройки. Куда нужно посмотреть и что изменить? Я подумал, что стоит взглянуть на Cinnamon, и установил его. Никаких проблем. Но я еще не решил до конца, что мне больше нравится, и хотел бы пресечь проблему с Unity, не перегружая свой домашний раздел.

Ян Дормер [Ian Dormer]

Это известная ошибка данной версии Unity. Ее обычное излечение – выполнение двух следующих команд:

```
unity --reset
unity --replace
```

Первая решает проблему, вторая перегружает Unity с новыми настройками. Они должны влиять только на версию 5.8, а текущая версия – 5.12, поэтому загрузка последнего релиза с Software Centre должна предотвратить возврат ошибки.

Если эти команды не помогают, может понадобиться очистка кэша Unity командой

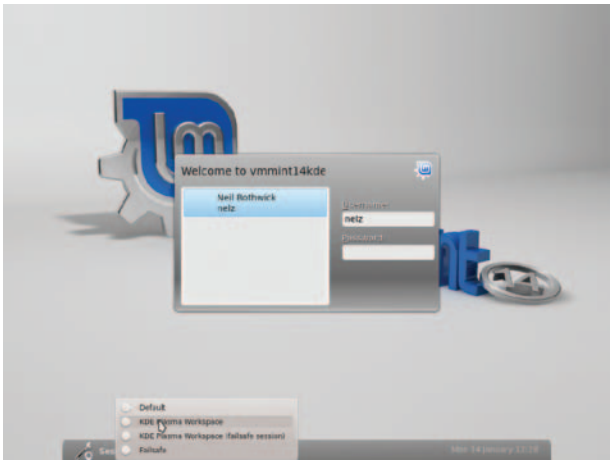
```
rm -rf ~/.cache/unity
```

Затем снова выполните две эти команды.

Также сообщалось, что эту проблему могут вызвать отсутствующие пакеты – убедитесь, что пакеты *unityplace-applications* и *unity-place-files* установлены и обновлены до последней версии. Наконец (как я надеюсь), похожие симптомы может вызвать то, что *zeitgeist* либо не установлен (маловероятно, если Вы не ковырялись в настройках APT), либо его база данных повреждена. Чтобы выяснить, не в этом ли дело, переименуйте базу данных и перезапустите Unity:

```
mv ~/.local/share/zeitgeist ~/.local/share/zeitgeist.bak
unity --replace
```

Если после этого проблема исчезла, можно удалить файл **zeitgeist.bak**.



› ***KDM*** **запоминает выбранную среду, но после установки другого оконного менеджера *KDM* прекращает действие.**

3 Грубая защита

Я попытался максимально защитить свой ноутбук — слишком хорошо знаю, как легко его потерять или пасть жертвой кражи. Я зашифровал свой домашний каталог с помощью опций в установщике Ubuntu и отключил загрузку по USB в BIOS, но что помешает кому-нибудь изменить настройки *Grub*, чтобы перейти к загрузке однопользовательского режима, или даже загрузить другое ядро с флэшки?

Майкл Ричардсон [Michael Richardson]

Ответ на Ваш вопрос – «ничто не помешает», по крайней мере, с настройками *Grub* по умолчанию. Однако можно настроить парольную защиту *Grub*, чтобы не все пункты меню запускались без пароля – для выбора определенных пунктов меню или их редактирования потребуются и пароль, и имя пользователя. Для начала задайте имя пользователя и пароль, создав файл **/etc/grub.d/00_security**, содержащий следующие четыре строки:

```
cat << EOF
set superusers="your_user"
password your_user your_pass
EOF
```

Имя файла имеет значение, так как оно влияет на то, где в окончательной конфигурации *Grub* появятся эти операторы. Вам могли бы посоветовать добавить эти строки в конец файла **00_header**, но этот файл может быть заменен при обновлении программы, и пароли будут удалены. А поскольку **00_security** будет считываться сразу после **00_header**, эти операторы попадут в то же самое место, но не будут перезаписаны. Имя пользователя можно выбрать любое – оно не обязательно имеет отношение к именам пользователей системы. Можно создать и других пользователей, но здесь сработает только суперпользователь. Если теперь выполнить команду **update-grub** и перезагрузиться, то для выбора любого пункта меню потребуется ввод имени пользователя и пароля. Это отличается от более ранних релизов *Grub*, когда неоговоренные пункты меню мог запускать кто угодно. Чтобы пункт меню могли загружать все, нужно добавить **--unrestricted** в строку **menuentry** в **grub.cfg**, но этот файл генерируется *update-grub* – вот мы и изменим скрипт, который его создает. Откройте **/etc/grub.d/10_linux** и замените

```
printf "menuentry '${title}' ${CLASS} {\n" "${os}" "${version}"
```

на

```
printf "menuentry '${title}' ${CLASS} --unrestricted {\n" "${os}" "${version}"
```

Меню загрузки *Grub* можно защитить, задав пароль на редактирование пунктов меню.

Чтобы сделать то же самое для пункта меню с Windows, измените **30_os-prober**. На этот раз мы применим **sed**, потому что изменений несколько:

```
sudo sed 's/--class os /--class os --users /' -i /etc/grub.d/30_os-prober
```

Помните, что ошибка в любом из этих файлов может привести к тому, что система перестанет загружаться, поэтому перед их изменением обязательно сделайте резервные копии (за пределами **/etc/grub.d**).

В качестве альтернативы можно добавить общедоступные пункты меню в файлы **40_custom** в **/etc/grub.d**, – в них можно точно указать необходимые опции. После этого запустите **update-grub** для повторной генерации меню, перезагрузите систему и проверьте меню.

Возможно, Вы заметили зияющую дыру в безопасности при таком подходе – пароль в двух местах в виде обычного текста. Мы сделали так для простоты тестирования и для того, чтобы было лучше понятно, что происходит. Когда все заработает так, как Вам нужно, запустите команду

```
grub-mkpasswd-pbkdf2
```

чтобы создать зашифрованную версию пароля, и замените строку про пароль в **00_security** на

```
password_pbkdf2 your_user encrypted_password
```

Снова запустите **update-grub**, и теперь у Вас есть эффективная и надежная защита.

4 Очистка кэша силой

В Проверив свертку MD5 для нескольких файлов на внешнем USB-диске, я обнаружил, что диск неисправен. Для одного файла — и только для него — значение свертки менялось с каждой попыткой. Свертка для остальных файлов оставалась постоянной. Я также запустил Memtest86+, чтобы исключить ошибки в оперативной памяти. Но мне пришлось чередовать попытки, чтобы избежать чтения из кэша.

Я сообразил, что при вычислении суммы командой md5sum **несколько раз подряд для одного** »

Терминалы и суперпользователи

Мы часто предлагаем в качестве решения проблемы ввести те или иные команды в терминале. Хотя обычно то же самое можно сделать с помощью графических утилит дистрибутива, такие решения будут слишком конкретными (будут зависеть от дистрибутива). Команды в терминале более гибкие и – самое главное – ими можно пользоваться во всех дистрибутивах. Команды настройки системы часто нужно выполнять от имени суперпользователя, называемого также root. Существует два основных способа это делать, в зависимости от используемого дистрибутива. Во многих, особенно в Ubuntu и его производных, перед командой можно написать **sudo** – при этом будет запрошен пароль пользователя, и ему будут предоставлены привилегии root только на время выполнения команды. В других дистрибутивах используется **su**, для использования которой требуется ввести пароль root и которая предоставляет полный доступ root до того момента, пока вы не наберете **logout**. Если в вашем дистрибутиве используется **su**, запустите ее один раз и выполняйте любые заданные команды без предшествующей **sudo**.

Коротко про...

Запуск нескольких команд

При работе в командной строке часто бывает нужно последовательно запустить несколько команд (такое бывает и в графических программах, но решение там далеко не столь простое). Классический пример – стандартный способ компиляции программ из исходников, когда нужно запустить **./configure** (возможно, с несколькими аргументами), затем **make**, а затем **make install**. На выполнение каждой команды может потребоваться от нескольких секунд до нескольких часов, в зависимости от сложности кода и скорости компьютера. Ждать, пока закончится одна команда, чтобы запустить другую, неудобно, поэтому можно сделать так:

```
./configure; make; make install
```

Точки с запятой означают, что команды будут выполняться друг за другом, как если бы вы запускали каждую отдельно. Некоторые из вас, возможно, уже заметили потенциальную проблему: что произойдет, если **./configure** или **make** завершится неудачно? Запустятся ли следующие команды в любом случае? Ответ – да, причем сообщения об ошибках в выводе будут скрыты, и вы даже не поймете, что ошибка была, пока не попытаетесь запустить программу. Более безопасный способ запустить эти команды таков:

```
./configure && make && make install
```

&& – логический оператор. Эта строка на самом деле означает: «если **./configure** верно и **make** верно и **make install** верно». К счастью для нас, оболочка сама определяет, «верна» ли команда – она запускает ее и смотрит, не вернула ли она ошибок. Если команда завершилась с ошибками, нет смысла запускать следующую, потому что условие уже ложно, поэтому при объединении операторами && команды запускаются друг за другом, но если какая-то команда завершилась неудачно и нам не надо больше сидеть над оболочкой, запуск команд прекращается. Другой оператор – ||, он означает «или». Так, в последовательности:

```
command1 || command2
```

команда **command2** будет запущена, только если **command1** завершится неудачно. Это не так полезно в интерактивной командной строке, но часто используется в скриптах, например, так:

```
somecommand1 || echo "Что-то поломалось!"
```

файла этот файл, следовательно, считывался из кэша, а не с диска. Это было совершенно ясно по скорости вычислений и из того факта, что не было звука перемещения головки диска (все файлы весили по несколько сотен МБ). Можно ли заставить md5sum **считывать файл с диска, очищая кэш?**

Питер Кристенсен [Peder Christensen]

Очистить дисковый кэш просто, но делать это нужно глобально, а не только для одного диска. Сделать это можно, записав данные в соответствующий «файл» в файловой системе **/proc**.

```
sync
sudo sh -c 'echo 3 >/proc/sys/vm/drop_caches'
```

Команда **sync** записывает на диски все данные, стоящие в очереди для записи на диски. Ее можно запустить от имени обычного пользователя, но в файл **/proc/sys/vm/drop_caches** разрешается писать только суперпользователю-root, поэтому довольно запутанная команда, которая помещает всю команду в строку и затем передает ее интерпретатору оболочки

```
sudo echo 3 >/proc/sys/vm/drop_caches
```

работать не будет. **Echo** запустится от имени root, а перенаправление вывода – от имени обычного пользователя, и команда завершится неудачно. Есть другой способ отправить вывод команды в файл от имени root:

```
echo 3 | sudo tee /proc/sys/vm/drop_caches
```

Прием с командой **tee** хорошо работает с более сложными командами, а для простой команды **echo** первый вариант проще, но оба достигают одинакового результата.

Контрольные суммы MD5 – не лучший способ проверки диска; для этого есть специализированные программы. Обычно используется *badblocks*, но сбойные блоки лучше искать с *e2fsck* (для файловых систем ext3/4). Команда

```
e2fsck -c /dev/sdb1
```

проверит заданный раздел на наличие сбойных блоков. При обнаружении таковых они будут добавлены в «черный список» файловой системы, чтобы в будущем они не употреблялись для файлов или каталогов. Существующим файлам это не поможет (хотя об этом позаботились Вы с проверкой контрольной суммы), но позволит предотвратить повреждение данных из-за ошибок на диске в будущем.

Другая подходящая программа – *smartctl*, она запрашивает данные S.M.A.R.T (Self-Monitoring, Analysis and Reporting Technology – технология самоконтроля, анализа и отчетности) большинства дисков. Бегло просмотреть статус диска можно командой

```
smartctl -a /dev/sdb
```

Обратите внимание на раздел "Errors logged [Обнаруженные ошибки]" в конце вывода **smartctl**. Заметьте, что *badblocks* и *e2fsck* запускаются для раздела, а *smartctl* – для всего диска. Быстро справиться о «здоровье» диска можно в любой момент, командой

```
smartctl -H /dev/sdb
```

Также можно настроить запуск *smartd* в виде демона с помощью менеджера запуска служб дистрибутива. Тогда он будет регулярно отслеживать состояние дисков и при обнаружении ошибок выдавать сообщение. В файле **/etc/smartd.conf** можно указать проверяемые диски и электронный адрес, куда будут отправляться оповещения. Файл подробно откомментирован.

5 Имена хостов в Android

У нас дома несколько устройств с Android — телефонов и планшетов, и есть проблемы с их идентификацией в сети. Дело в том, что имя каждого устройства начинается с "android", затем следует длинная шестнадцатеричная строка. У всех остальных устройств, даже у шального устройства Apple, затесавшегося в нашу сеть, осмысленные имена хостов.

Я покопался в настройках устройств Android, на всех из них запущены Ice Cream Sandwich или Jelly Bean. Я нашел способ изменить имя устройства при подключении по Bluetooth, но не имя хоста, используемое для беспроводного подключения.

Фил Уотсон [Phil Watson]

Вы не можете найти настройку потому, что ее там нет. Однако не все потеряно – хорошая новость в том, что имя хоста устройства Android можно изменить; потенциально плохая – в том, что сделать это можно только на устройстве с root-доступом. Вдобавок придется работать в командной строке, поэтому для начала нужно установить терминал – я пользуюсь Android Terminal Emulator.

Откройте терминал, затем выполните две такие команды:

```
su
getprop net.hostname
```

При этом должно вывестись текущее недружелюбное имя. Чтобы изменить его, скомандуйте

```
setprop net.hostname yourchosenhostname
```

Нужно указывать только имя хоста, а не полное доменное имя, т.е. **myphone**, а не **myphone.my.network**. Это изменение постоянное, по крайней мере, до следующего запуска **setprop**; оно сохраняется после перезагрузки. Насколько мы знаем, сделать это на устройствах без доступа root нельзя, но получить root-доступ к большинству устройств Android нынче очень просто.

Если на устройстве Android уже установлен сервер SSH, такой как *QuickSSHd*, можно запускать команды по SSH, подключившись со своего компьютера, вместо установки эмулятора терминала на телефон.

6 Управление кулером

В моем компьютере с креативным названием LinuxBox3 установлена материнская плата Asus P8H6-MX с автоматическим управлением кулером. Однако настройки BIOS не оказывают на кулер никакого влияния, и, может быть, есть какой-то драйвер для MS Win, который и делает всю работу?

Я пользуюсь Fedora 16 и на LinuxBox3, и на своем нетбуке Acer Aspire One. Странно, но кулер на ноутбуке ускоряется и замедляется в соответствии с загрузкой процессора, поэтому проблема, вероятно, существует только для материнской платы Asus. Есть ли модуль или утилита, которую можно установить, чтобы воспользоваться этой функцией?

Кен Уилсон [Ken Wilson]

Похоже, что для управления вентиляторами на этой материнской плате у Вас должны быть загружены три модуля: *coretemp*, *pkgtemp* и *w83627ehf*. Судя по Вашему отчету HardInfo, модуль *pkgtemp* не загружен. Это объяснило бы то, что скорость всех вентиляторов, кроме одного (вероятно, вентилятора процессорного кулера), показывается равной нулю. Не видя скорости вентилятора, система управлять им не может, поэтому первый шаг – добавить *pkgtemp* в **/etc/modules**, чтобы модуль принудительно загружался при загрузке системы.

Также, возможно, потребуется добавить **acpi_enforce_resources=lax** к опциям ядра в Grub. Откройте **/etc/default/grub** и добавьте эту опцию без кавычек в конец переменных **GRUB_CMDLINE_LINUX_DEFAULT** и **GRUB_CMDLINE_LINUX**. Выполните команду

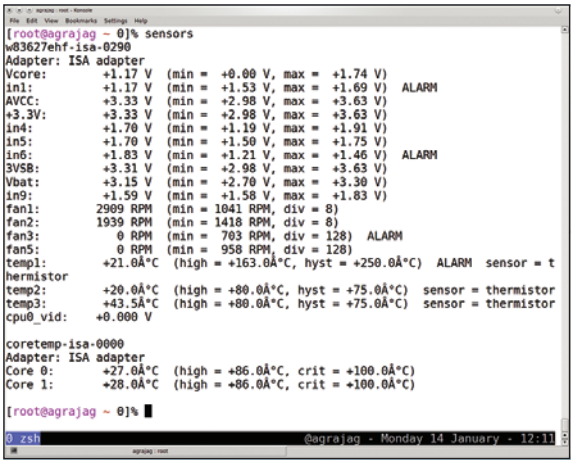
```
sudo grub-mkconfig -o /boot/grub2/grub.cfg
```

для обновления опций загрузки и перезагрузите систему. Теперь Вы должны увидеть скорость вращения вентиляторов в выводе команды

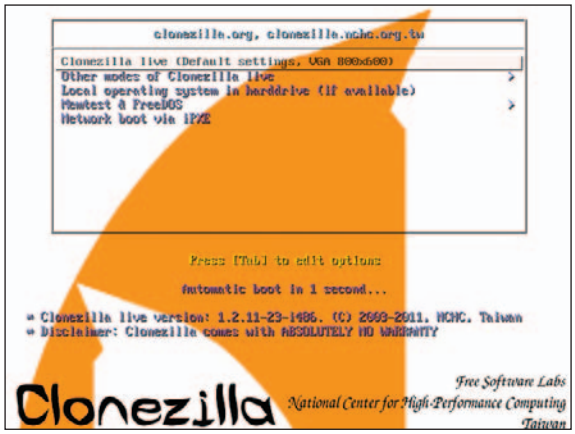
```
sensors
```

Когда скорость начнет отображаться, вентиляторами можно будет управлять с помощью команд *pwmconfig*, *fancontrol* (обе – из пакета *lm_sensors*) или любой графической программы.

Экспериментировать с настройками вентилятора кулера следует при минимальной загрузке системы. Если Вы по ошибке снизите скорость вращения вентилятора, в этот момент лучше не компилировать ядро и не кодировать видео



Скорость вращения вентиляторов в кулерах может просмотреть *lm_sensors*, а управляются они из системы путем загрузки соответствующих модулей.



Clonezilla — это хорошо, но перед созданием резервной копии убедитесь, что на диске нет ошибок.

1080p. В частности, команда **pwmconfig** в рамках своих тестов может на несколько секунд выключить вентиляторы совсем.

6 Странный диск

Я регулярно копирую весь жесткий диск на внешний жесткий диск в Ubuntu 12.04 с помощью Clonezilla. Каждый раз, когда я прошу Clonezilla проверить систему перед созданием образа, я получаю следующее сообщение:

```
Free blocks count is wrong. Fix <y> ?
Free inodes count is wrong. Fix <y> ?
```

[Подсчет свободных блоков неверен. Исправить? Подсчет свободных узлов неверен. Исправить?]

Я отвечаю «y», и все вроде нормально. Это ошибка в программе или я делаю что-то не так?

Брайан Митчелл [Bryan Mitchell]

Первым делом нужно проверить, проявляется ли эта проблема только в Clonezilla. Выключите компьютер как обычно, затем загрузитесь с любого Live CD или DVD, который есть под рукой – прекрасно подойдет установочный диск Ubuntu – откройте терминал и выполните команду

```
sudo fsck -f /dev/sdaN
```

Список таблиц разделов можно получить командой

```
sudo fdisk -l
```

Если здесь ошибок в файловой системе нет, проблема связана с Clonezilla, и о ней лучше сообщить в рассылку или на форумах Clonezilla – ссылки можно найти на http://clonezilla.org. Если с Live CD появляются те же самые ошибки, есть несколько вариантов.

Возможно, какая-то программа не завершается корректно, что приводит к некорректному размонтированию файловой системы. Вам нужно посмотреть сообщения при выключении системы – нажмите E в меню *Grub* при загрузке системы и удалите **quiet splash** из параметров ядра, затем нажмите Ctrl+X для загрузки. Тогда будут показаны все сообщения при загрузке системы и, что более важно, при последующем выключении системы – поищите в них что-либо подозрительное.

Причина может быть и в ошибках жесткого диска – проверьте его командой **smartctl** (подробности см. в ответе на один из предыдущих вопросов – «Очистка кэша силой»). LXF

Помогите нам помочь вам

Ежемесячно мы получаем несколько писем, на которые не в состоянии ответить, так как проблема описана в них недостаточно полно. Чтобы дать вам наилучший ответ, нам нужно знать как можно больше.

Если у вас появляется сообщение об ошибке, приведите его точный текст и опишите конкретные условия, когда оно появляется. При возникновении проблемы с устройствами перечислите нам все установленные устройства.

Если Linux уже запущен, можете применить для этого отличную программу *Hardinfo* (http://hardinfo.berlios.de/) – она сохранит подробную информацию об устройствах и о состоянии системы в HTML-файл, который вы сможете приложить к своему письму.

Альтернативный и не менее удобный вариант – *lshw* (http://ezix.org/project/wiki/HardwareLiSter). Одна из указанных программ непременно должна быть включена в ваш дистрибутив (а иногда 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

Руткиты

Что такое руткит?

Руткит – набор программ, устанавливаемый взломщиком, чтобы скрыть свое присутствие на компьютере. Он включает, например, модифицированные версии **ls** и **ps**, которые не показывают файлов и процессов, принадлежащих взломщику. Чтобы снизить подозрения, такие программы обычно представляют собой трояны. Название объясняется тем, что они устанавливаются и могут запускаться от имени пользователя root. Пораженный компьютер обычно называется “rooted”.

Какой ущерб системе обычно наносит руткит?

Сам по себе не такой большой. Его задача – дать злоумышленнику путь к вашему компьютеру так, чтобы у вас не возникло подозрений. Для этого он в основном изменяет программы для отслеживания состояния системы, чтобы они не замечали ничего необычного. Этим руткиты отличаются от троянов.

А что такое «троян»?

Троянский конь – часто сокращается до «троян» – вредоносная программа, замаскированная под нормальную. Названием он обязан коню, с помощью которого греки проникли в Трою – отсюда фраза «бойтесь данайцев, дары приносящих, и сайтов с пиратскими программами».

Пример трояна – к счастью не представляющего большой проблемы для пользователей Linux – программы с рекламой. К ним же относятся и другие программы, которые «звонят домой» без вашего ведома и согласия.

Имеет ли это отношение к музыкальным компакт-дискам?

Только то, что последнее ПО DRM от Sony описывается как руткит, хотя и для Windows. Тот факт, что оно устанавливается столь известной компанией на такое количество компьютеров без уведомления и без согласия пользователей, делает ее самым откровенным (на данный момент) руткитом.

Как узнать, поражена ли система руткитом?

Существуют различные программы для проверки системы на признаки наличия руткита. Они работают двумя основными способами – путем поиска известных последовательностей кода руткитов и путем сравнения контрольных сумм различных файлов с контрольными суммами этих файлов в «чистой» системе. Из-за последнего факта важно установить и запустить программу как можно раньше после установки системы, чтобы она записала контрольные суммы для «чистой» системы. Две таких программы – *Rootkit Hunter* с www.rootkit.nl и *chkrootkit* с www.chkrootkit.org. Их надо запускать регулярно, лучше из *cron*. Хорошо, если компьютер будет подключен к Интернету, чтобы они автоматически обновляли свои базы данных известных руткитов.

Вылечат ли систему эти программы?

Нет. Установка на компьютере руткита серьезно компрометирует систему. Руткит помогает взломщику замести следы своих действий, так что эти действия могли затронуть и любые другие файлы. Безопасный выход один: сделать резервную копию своих данных и файлов настройки и переустановить систему.

LXFHotPicks

FatRat » Avogadro » Safe-rm » XOWA » Cppcheck » Parcellite » PokerTH » Erebus » Gtkdialog » LiteIDE » Mednafen

Майк Сондерс
Излазив самые недосягаемые и укромные уголки Интернета, Майк точно знает, где прячутся лучшие образчики открытого кода.

Менеджер закачек на стероидах

FatRat

Версия: 1.2b2 **Сайт:** http://fatrat.dolezel.info

FatRat – Большой Папа всех менеджеров закачек. Это – петух в курятнике и царь горы; бьемся об заклад на фунт стерлингов, что им пользуется Чак Норрис. Обычно менеджеры закачек весьма прозаичны – пыхтят себе помаленьку в фоновом режиме, пока вы занимаетесь делом. Однако *FatRat* просто ошарашил нас огромным выбором функций.

Написанный на C++, с интерфейсом на *Qt4*, *FatRat* – «универсальный менеджер закачек»: он умеет скачивать файлы из самых разных источников и протоколов, а не просто через сайты, как многие ему подобные. Среди этих источников и протоколов – HTTP, HTTPS, FTP и BitTorrent. Программа также включает функции загрузки: вы можете заливать файлы на FTP-серверы и создавать файлы **.torrent**, не прибегая к отдельным инструментам.

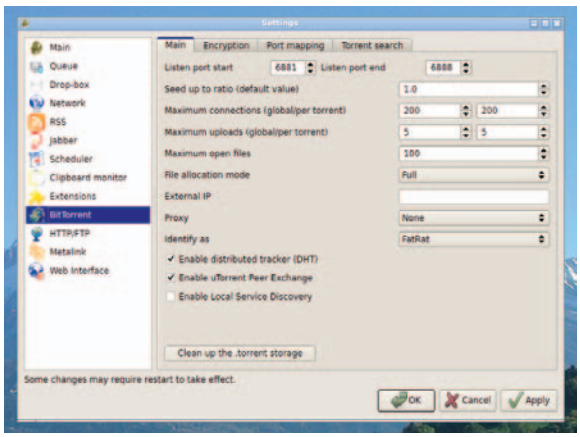
> «FatRat умеет скачивать файлы из разных источников.»

Чтобы скачать файл, можно либо перетащить ссылку на него из окна браузера в панель Transfers в *FatRat*, либо щелкнуть по кнопке Add и вставить URL. Последний способ позволяет вставить несколько URL для одновременной закачки нескольких файлов. По умолчанию эти закачки хранятся в Main Queue [Главная очередь], и здесь-то и начинает проявляться во всем блеске мощь *FatRat*. Вы можете создавать несколько очередей закачки с разными скоростями закачки, выставляя таким образом приоритетность передачи файлов, а это бывает невероятно удобно.

Например, вам может понадобиться установить очередь закачки ISO каких-нибудь дистрибутивов с низким приоритетом, поскольку вы намерены заняться ими только через несколько дней, и выставить высокий приоритет с более высокой скоростью для того, что требуется срочно.

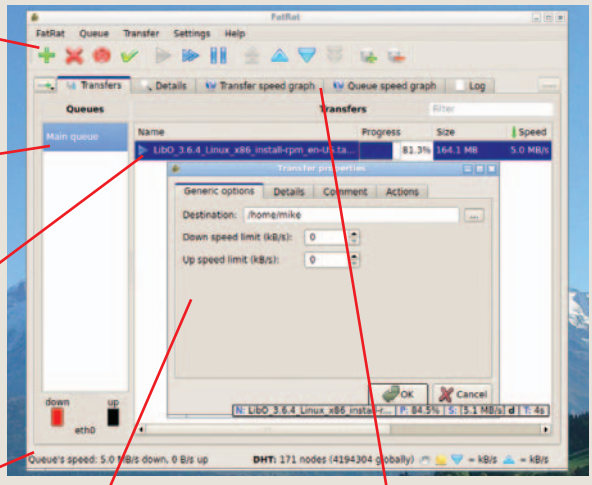
Еще имеются графики отдельных передач и очередей, подробная информация об отдельных закачках, журнал активности программы и дополнительные функции, такие, как возможность запуска определенной команды после завершения закачки. Мы не можем придумать ни одной разумной функции, относящейся к закачке, которой бы не было в *FatRat*, а экран Settings [Настройки] настолько подробен, что можно потратить несколько дней на его изучение.

Если рудиментарные инструменты закачки, встроенные в ваш браузер, больше не оправдывают ваших надежд, вы просто обязаны попробовать этот. И стать, как Чак Норрис (а вдруг!).



› *FatRat* до такой степени настраиваем, что его окно Settings можно изучать годами.

Исследуем интерфейс FatRat



Управление
Добавляйте, удаляйте, ставьте на паузу и изменяйте порядок закачек с помощью этих кнопок.

Очереди
У вас может быть несколько очередей с разными скоростями – опции можно увидеть в меню Queue.

Передачи
Этот список отображает все активные и поставленные на паузу закачки для текущей очереди. Щелкнув правой кнопкой, вы увидите свойства.

Состояние
Эта строка показывает текущую скорость закачки и раздачи. Щелкните дважды по голубой кнопке, чтобы изменить границы...

Свойства
...и вы увидите это окно для настройки скорости и добавления действий после завершения закачки.

Вкладки
Здесь можно переключаться между основным видом Downloads, отдельными деталями, графиками скорости и журналом.

Редактор молекул

Avogadro

Версия: 1.1.0 **Сайт:** http://avogadro.openmolecules.net

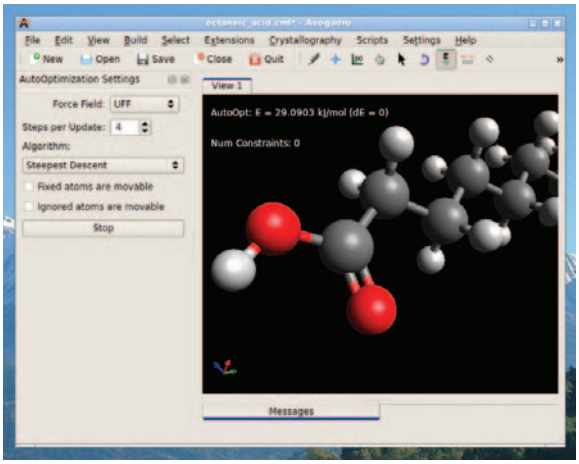
В школе мы не испытывали особенно пылкой любви к урокам химии. Нет, без сжигания домашних заданий или растворения линейки в соляной кислоте не обходилось – но всю невероятную важность атомов мы вполне осознали уже во взрослой жизни. *Avogadro* – редактор молекул, и одна из тех программ, которые превосходно демонстрируют достоинства открытых разработок: при всей узости ее целевой ниши, она продолжает существовать, потому что не испытывает коммерческого давления.

Ее приложения весьма распространены, в том числе в вычислительную химию, материаловедение и биоинформатику. Многие дистрибутивы включают *Avogadro* в свои репозитории, но если в вашем дистрибутиве нет последней версии или вы хотите скомпилировать ее из исходника, чтобы получить самые современные функции, вам сначала придется установить *Qt*, *Eigen2* и *OpenBabel* в качестве основных зависимостей.

Подготовив ее к работе и запустив, вы сможете начать создавать молекулы, выбрав инструмент Draw – значок руки. Хотя интерфейс минимален, при наведении мыши на кнопки появляются удобные подсказки. Затем можете выбирать элементы из списка слева, даже выводить таблицу Менделеева, если хотите выбрать что-то нетривиальное. Используя инструмент Navigation, вы сможете поворачивать, менять положение и изменять масштаб, а инструмент Manipulation изменяет положение атомов и связей между ними.

Avogadro поддерживает *OpenBabel*, позволяя ему импортировать большое количество форматов химических файлов, и для начала включено несколько фрагментов (**/usr/share/avogadro/fragments/**).

> «Впечатляюще сделанный инструмент с хорошим фокусом.»



› Предпросмотр грядущего официального апокалиптического устройства от LXF. Дурачье! Мы им покажем...

Благодаря поддержке Python вы можете создавать скрипты внутри приложения, а система расширений позволяет нашпиговать приложение добавочными функциями. Документация довольно бессистемная; онлайн имеются различные текстовые и видеоруководства, однако некоторые функции программы имеют либо очень скудную информацию, либо не имеют ее вообще.

В общем и целом это очень впечатляюще сделанный инструмент с хорошим фокусом на определенной области, и ровный рендеринг OpenGL и анимация превращают работу в нем в истинное удовольствие.

Инструмент безопасности файловой системы

Safe-rm

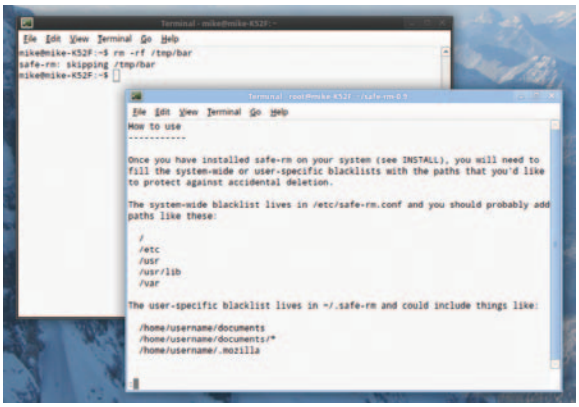
Версия: 0.9 **Сайт:** www.safe-rm.org.nz

Неловкая ситуация № 261: к вам пришел друг – пользователь Windows, и вы ему демонстрируете величие Linux. Вы хотите показать ему мощь командной строки и то, как пара нажатий клавиш заменяет многочасовое щелканье мышью в менеджере задач. Все идет замечательно, а потом... упс, вы нечаянно удаляете **/usr**. И в ужасе наблюдаете, как ваша система Linux на глазах распадается на куски, а ваш друг в это время самодовольно ухмыляется. «Я-то думал, эта ваша Линуха надежна, – злорадствует он, пока вы лихорадочно пытаетесь собрать свою систему. – Похоже, эта ваша командная строка слишком уж мощная, а?» – ехидно добавляет он.

Оболочке подвластна ужасающая мощь, особенно если вы работаете от имени root. Затесался лишний пробел, обдернулись с флажком – и начинаются ужасы. *Safe-rm* нужен, чтобы обеспечить прослойку безопасности между клавиатурой и файловой системой, благодаря перехвату команд **rm** и неудалению файлов, заранее занесенных в черный список. Будьте уверены, вы не сможете сделать **rm -rf /usr** или удалить другие критически важные файлы – даже от имени root.

Работает это так: вы помещаете скрипт *safe-rm* куда-нибудь в $PATH и переименовываете его в *rm*. Помните: вы не должны переписывать бинарник *rm*, как это требуется. Поиск по $PATH осуществляется слева направо, и если **/usr/local/bin** идет в вашей системе перед **/bin**, поместите его в первую директорию. Когда вы введете **rm**, Linux сначала запустит «безопасную» версию в **/usr/local/bin**, а та затем вызовет **/bin/rm**. Теперь создайте **/etc/safe-rm.**

> «Это простое решение при серьезном администрировании.»



› *Safe-rm* рекомендует заблокировать директории /etc, /usr **и** /var **по умолчанию — это воистину мудрое решение.**

conf и добавьте пути, которые вы не хотите удалять ни при каких обстоятельствах, по принципу один на строку. Если, например, вы добавите **/usr** и в оболочке введете **rm -rf /usr**, вы увидите

```
safe-rm: skipping /usr
```

Личные черные списки для каждого пользователя можно создать в **.safe-rm** в каждой домашней директории. Это простое решение, когда вы при серьезном администрировании не хотите случайно грохнуть важную директорию.

Программа для просмотра Wikipedia оффлайн

XOWA

Версия: 0.2.0.1 **Сайт:** http://bit.ly/X5YEih

› ***XOWA*** **— приложение Java, поэтому легко устанавливается и работает на большинстве основных ОС.**

За всей болтовней об облачном хранении, потоках видео по запросу и мобильных устройствах всегда в сети, легко забыть, что многие люди в этом мире отнюдь не обладают надежным или быстрым интернет-соединением. Благодаря обратной связи от читателей мы знаем, что **LXFDVD** очень популярны в отдаленных местностях, где dial-up – обычное дело, а электроэнергию отключают что ни день. И мы сейчас говорим не о странах третьего мира – многие сельские районы Англии по-прежнему маются с допотопной интернет-структурой.

Однако в любом случае *XOWA* – это приложение, позволяющее работать с Wikipedia оффлайн. Зачем это надо? Ну, представьте себе, что вы оказались в одном из вышеописанных мест, с ограниченным доступом к Интернет. Вам нужно донести все прелести Wikipedia до школы или библиотеки, да так, чтобы не приходилось заниматься подключением при каждом поисковом запросе.

С помощью *XOWA* вы загрузите всю базу данных Wikipedia одним махом, и затем будете использовать ее оффлайн по своему усмотрению. Это эффективный оффлайн-браузер, который скачивает моментальные снимки с Wikipedia и соответствующих сайтов, таких, как Wiktionary и Wikiquote. По умолчанию он качает текстовую версию English Simple Wikipedia (около 375 МБ), и можно также добавить изображения. Конечно, можно загрузить хоть всю Wikipedia на разных языках, но для этого понадобится несколько сотен гигабайт на жестком диске.

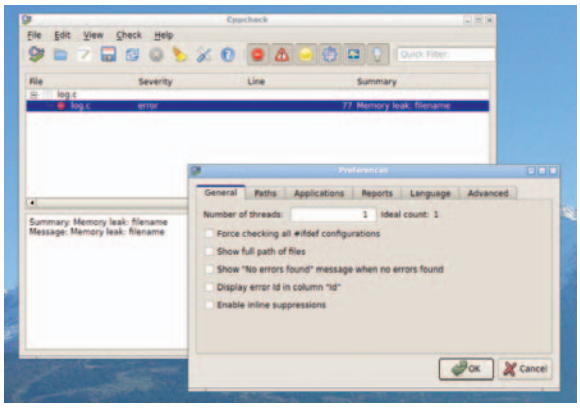
XOWA позволяет и редактировать статьи; здесь нет очевидного способа заливать ваши изменения назад в Wikipedia, но в будущем это обещает стать отличной функцией. Вы скачиваете мгновенный снимок, делаете свои изменения оффлайн и заливаете их обратно, получив соединение с Интернет. В общем и целом, это простое приложение, которое не пытается делать что-то особенное, но очень помогает в местах с ограниченным доступом к Сети. Wikipedia – истинная сокровищница информации, и чем больше людей смогут использовать ее – независимо от надежности или скорости их интернет-соединения – тем лучше. И если вы планируете провести отпуск в какой-нибудь затерянной лаосской деревушке, где Интернета и духу нет, почему бы вам не прихватить с собой старушку Wikipedia?

> «Скачивает текстовую версию English Simple Wikipedia.»

Анализатор исходного кода C/C++

Cppcheck

Версия: 1.58 **Сайт:** http://cppcheck.sourceforge.net



› ***Cppcheck*** **лучше всего работает в составе IDE, но и его собственный интерфейс вполне достойный.**

Идеальных людей нет. Даже если вы относитесь к мировой элите программистов C/C++, и девушки при виде вашего исходного кода млеют, все равно вы не застрахованы от ошибок. Вас может утомить работа над какой-то частью кода, или в некой зоне серого вещества в вашем черепе произойдет сбой памяти на пару секунд. Какой бы ни была причина, ошибки случаются, а даже самая мелкая опечатка в исходном файле C/C++ может иметь катастрофические последствия.

Cppcheck – это инструмент статического анализа для вышеупомянутых языков: он не открывает Америк, а просто просматривает ваш исходный код, пытаясь выявить ошибки и потенциальные проблемные фрагменты, которые не выявляет *GCC* (даже при включенном **-Wall**). Можете запускать его прямо из командной строки или интегрировать в разные IDE и редакторы, такие, как *Eclipse*, *Code::Blocks* и *gedit*. Есть также и GUI-версия, доступная через **cppcheck-gui**, которая добавит удобства в работе, позволив работать мышью.

Какой бы способ вы ни предпочли, *Cppcheck* может определить проблему с утечками памяти; выходом за границы массива; исключением безопасности; использованием устаревших функций; недействительным использованием Standard Template Library; и со множеством мелочей по части стиля кода и производительности. Результат получается немногословным, но вполне понятным, и если вы используете *Cppcheck* в большом проекте и хотите включить результат в формат готового отчета, вы можете создать файл XML из полученной информации.

Cppcheck хорошо документирован и предлагает достаточно подробную информацию в справочнике и в wiki. Можно даже создать индивидуальные правила (в XML) для добавления проверок, которых нет в программе по умолчанию, но которые могут оказаться уместными для вашего кода.

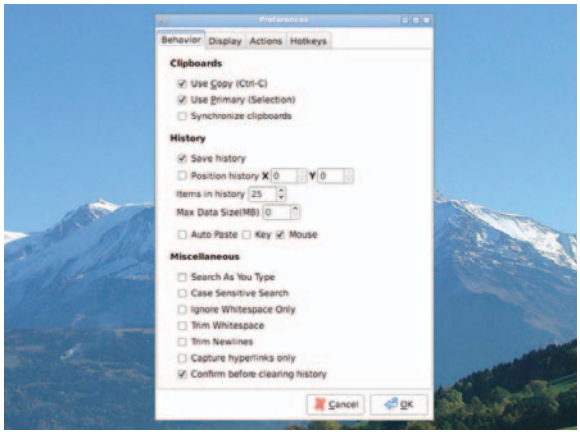
В общем и целом, это хорошо продуманная, быстрая и удобная программа для тех, кому приходится много работать в C/C++ и хочется сгладить все шероховатости кода, прежде чем отправить его в большой мир.

> «Можно создать правила, которых нет по умолчанию.»

Менеджер буфера обмена

Parcellite

Версия: 1.1.3 **Сайт:** http://parcellite.sourceforge.net



› Можете сохранять журнал использования буфера обмена и даже экспортировать его.

Смеха ради попробуйте сделать так: откройте *AbiWord*, наберите предложение, потом скопируйте его в буфер обмена, выделив и нажав Ctrl+C. Теперь нажмите Ctrl+V, и вы его вставите – обычное дело, да? Теперь наберите другое предложение, выделите его и щелкните где-нибудь средней кнопкой мыши (или используйте обе кнопки трекпада, если у вас ноутбук). Вы увидите, что щелчок средней кнопкой вставил выделенный текст. Что же тут происходит? У вас работают два буфера обмена, и этим мы обязаны восхитительно гибкой (и иногда жутко сбивающей с толку) природе *X Window System*. Буфер обмена средней кнопки управляется *X*, низкоуровневым графическим слоем, а Ctrl+C/V управляется *GTK* – высокоуровневым инструментарием.

Менеджер буфера обмена вроде *Parcellite* делает для *GTK* ситуацию более понятной. Запустив его, вы увидите небольшой значок с буквой P в системном лотке – по щелчку левой кнопкой перед вами появится всплывающее меню со списком пунктов, которые вы недавно скопировали в буфер обмена (если там пусто, заполните его текстом, скопированным с web-страниц и документов). Теперь, когда бы вы ни щелкнули по пункту в меню, он станет текущим содержимым буфера обмена.

Однако по-настоящему актуальным *Parcellite* становится благодаря своей способности унифицировать две разных системы буфера обмена. Щелкните правой кнопкой по значку P в системном лотке и выберите Preferences. Затем в верхней части появившегося окна выберите опцию Use Primary (Selection) и нажмите OK. Теперь *Parcellite* будет не просто сохранять содержимое буфера обмена, сохраненное с помощью Ctrl+C, но также и то, что создано *X Window System* – например, то, что появилось благодаря выделению текста. Оно появится в списке *Parcellite*, выделенное курсивом.

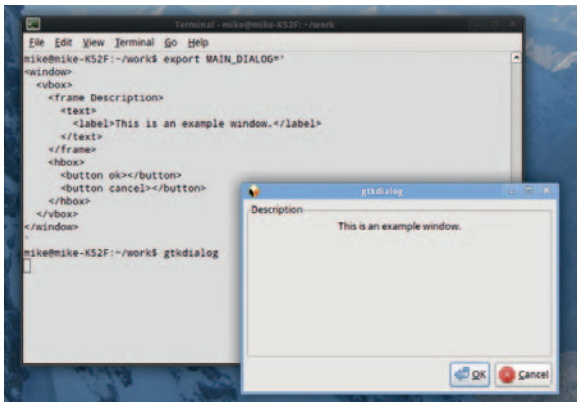
> «Parcellite делает для Gtk ситуацию более понятной.»

Мы всегда запускаем приложения смесью разных инструментариев, и разрозненная модель буфера обмена периодически немало нас огорчает, так что стали самыми преданными сторонниками *Parcellite*. Да, есть и другие менеджеры буфера обмена, но они привязаны к крупным средам рабочего стола, а это приложение имеет минимальные требования и отлично работает на легковесных менеджерах окон.

Инструменты GUI для скриптов

Gtkdialog

Версия: 0.8.3 **Сайт:** http://bit.ly/xHtKRy



› Простой пример – см. http://tinyurl.com/avueawk, где вы найдете созданные пользователями скрипты с уймой кнопок и выпадающих списков.

Штатный ли вы программист или только балуетесь, все равно есть вероятность, что вы используете языки скриптов. На овладение основными навыками написания достойных программ не требуется особо много времени; но что если нужно добавить функции в GUI? Возможно, зарывшись в груду документации по *PyGTK*, вы все же разберетесь в его многочисленных функциях – но это бывает явным перебором. Если на данный момент вам просто надо, чтобы у программы появилось всплывающее диалоговое окно, зачем тут весь инструментарий целиком?

Gtkdialog – небольшое приложение, которое позволяет создавать диалоговые окна GUI из ваших скриптов. Вам не нужно переживать по поводу библиотек, или модулей, или ссылок; просто возьмите данные XML для описания диалогового окна в переменной среды, и *Gtkdialog* доделает остальное. Например, в оболочке:

```
export TEST_DIALOG="<text><label>Hello world</label></text>"
gtkdialog -program TEST_DIALOG
```

Вот так все просто – единственная текстовая строка отображается в окне с помощью кнопки Close. Но с помощью *Gtkdialog* вы можете добавлять кнопки, окна для ввода текста, флажки и изображения. Вы даже можете располагать виджеты горизонтально и вертикально. Можно создавать довольно сложные диалоговые окна, которые выглядят, как будто они появились из «родных» приложений *GTK*, и при этом не увязнуть в API *Gtk*.

Иногда бывает нужно получить информацию из диалоговых окон. *Gtkdialog* направляет результат в **stdout**, и если у вас диалоговое окно с кнопками OK и Cancel и вы щелкаете по Cancel, то увидите:

```
EXIT="Cancel"
```

> «Можно добавлять кнопки, окна, флажки и изображения.»

Вы можете сохранить это способом, наилучшим для своего языка программирования. Например, в *Bash* быстрым способом будет такой:

```
RESULT="`gtkdialog --program TEST_DIALOG`"
```

Есть и более сложные и элегантные методы обработки с использованием цикла **for** и **eval**. Это означает, что каждый виджет становится собственной переменной среды, поэтому для **EXIT="Cancel" string** вы получаете переменную среды **$EXIT**, которая содержит "Cancel".

HotGames Развлекательные приложения

Покер

PokerTH

Версия: 1.0 **Сайт:** www.pokerth.net

Для игры, где настолько важен фактор везения, просто удивительно, сколько книг написано о покере. Конечно, хороший игрок всегда обыграет плохого, но нам просто интересно, есть ли в конечном итоге хоть какая-то польза от всех этих руководств. Вопрос спорный: здесь важно умение держать себя в руках и быстро просчитывать вероятности. Однако игра набирает популярность, и мы рады видеть клиент коммерческого уровня для Linux.

Давние читатели **LXF** должны помнить, что мы уже рассказывали о *PokerTH* в **LXF143**, но он развивается гигантскими темпами. Для начинающих: интерфейс был переделан, и теперь он соответствует платным сайтам, и подобным же образом был перезапущен сайт, где теперь работает клиент Android. Он явно расцвел – как в однопользовательском режиме, так и для игры между людьми онлайн.

PokerTH отличается очень высокой степенью настраиваемости, с различными темами для игорных столов и высоким уровнем контроля за сетевыми играми (тайм-аут для игрока, стартовая наличность, частота повышения ставок, и т.д.). Вы можете включить журнал записи комбинаций – это удобно для ведения статистики – и включить панель «шанса», которая показывает вероятность получения определенных комбинаций в вашей текущей сдаче. Есть специальный сервер *PokerTH*, где играют на виртуальные деньги, но нам интересно будет узнать, попадет ли когда-нибудь *PokerTH* на коммерческие сайты.

> «Есть сервер *PokerTH*, где играют на виртуальные деньги.»

Из-за технических ограничений *PokerTH* не может воссоздать лучшие стратегии покера, такие, как чихание на фишки своего противника.

Многие игроки онлайн ворчат по поводу программ, жалуясь, что те мошенничают в чью-то чужую пользу или пытаются обобрать их (можно подумать, что у сайтов картежных игр бывают какие-то другие цели). Однако на сайте покера с открытым кодом при игре на настоящие деньги никто не будет в состоянии выдвигать подобные возражения, и придется сфокусироваться на собственном мастерстве.

Могут возникнуть проблемы, связанные с модифицированными клиентами, вследствие самой природы открытого кода, но нам бы чрезвычайно хотелось увидеть такой сайт в действии. Авось когда-нибудь и увидим.

RPG «наведи-и-щелкни»

Erebus

Версия: 0.4 **Сайт:** http://erebusrpg.sourceforge.net

Выпускайте релизы раньше, выпускайте релизы чаще – это мантра многих разработчиков открытого кода. Хотя нередко в ней есть смысл, особенно в контексте серьезных приложений, где нужна солидная обратная связь по дизайну до фазы плотного программирования, все же нам кажется, что в играх с открытым кодом ею злоупотребляют. Слишком часто приходится видеть объявления о выходе потрясающей новой игры, в которой релиз 0.1, как правило, просто эквивалентен сообщению «Добро пожаловать в мою игру». *Erebus* – не тот случай; пока что он находится на ранней стадии разработки, но в него уже вполне можно играть, и цели у него реально достижимые.

Вкратце, *Erebus* – это классическая ролевая бродилка по подземельям типа наведи-и-щелкни; вы выбираете своего персонажа и уровень сложности, а затем беретесь за дело, сражаясь с гоблинами, орками и зомби, не забывая собирать предметы и упиваться ростом набираемых очков. На экране заставки выберите, кем хотите быть – Варваром, Эльфом, Полурослико, Рейнджером или Воином, и нажмите на Start, чтобы погрузиться в игру. Затем щелкните для указания своему персонажу направления движения, и щелкайте по другим персонажам, чтобы с ними сразиться. Внизу слева показан доступ к своим предметам и статистике, а в самом низу – кнопки с «+» и «-» для масштабирования карты

> «Erebus — классическая бродилка типа наведи-и-щелкни.»

***Erebus* разработан с оглядкой на тачскрин, и скоро появится на вашем планшетнике.**

подземелья, и кнопка O, центрирующая камеру на вашей персоне. Пробел отвечает за разнообразные действия: например, разговоры с другими персонажами, сбор предметов и открытие дверей.

На данный момент в *Erebus* два квеста, и в плане презентабельности он пока что не сияет, но восхищают амбиции разработчика. Он уже выдал версию для телефонов Nokia Symbian с Android с заделом на будущее (когда будет закончена версия игры 1.0). Он также ищет музыкантов и художников, готовых заняться графикой, и если у вас есть творческая жилка и вы всегда мечтали помочь в создании игры с открытым кодом, вот вам шанс!

Среда разработки Go

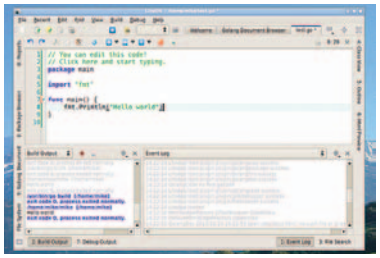
LiteIDE

Версия: 16 **Сайт:** http://code.google.com/p/liteide

Похоже, в наши дни Google умудрился запустить лапу в каждый пирог: поиск, электронная почта, социальные сети, карты и даже языки программирования в виде Go. Если вы пока не сталкивались с Go, то, вероятно, цинично вздохнете: «А миру нужен еще один язык программирования?» И отчасти будете правы, но Go создавали корифеи – Роб Пайк [Rob Pike] и Кен Томпсон [Ken Thompson], не один десяток лет занимавшиеся подобными вещами. С введением в Go вы можете познакомиться в **LXF150**.

LiteIDE соответствует своему названию: это легковесная среда разработки для редактирования кода Go. Главные ее зависимости – *Qt* и *QtWebKit*, которые сегодня являются вполне обычным делом в популярных дистрибутивах. Среди его основных функций – выделение синтаксиса, свертывание кода, редактор автозаполнения кода со встроенными частями IDE; вы сможете создавать проекты Go и просматривать ошибки компиляции и результаты программы, не выходя в оболочку. Интерфейс хорошо поддается индивидуальной настройке, его верхняя панель служит редактором кода, нижняя левая – для вывода результата компиляции, а нижняя правая – журнал событий.

Справочной информации пока немного, но документацию по Go покажет встроенный браузер документа. Это не огромная, на все способная IDE типа *Eclipse*, однако она вновь разожгла наш интерес к Go. Пожалуй, мы даже портируем ее в MikeOS...



› *LiteIDE* довольно минималистская по сравнению с другими приложениями и может использоваться для программирования на ходу с нетбука.

Многосистемный эмулятор

Mednafen

Версия: 0.9.28 **Сайт:** http://mednafen.sourceforge.net

Без эмуляторов нам работалось бы намного продуктивнее. Мы выполняли бы на современных компьютерах и программах свои задачи, вместо игры в *Stunt Car Racer* на виртуальной Speccy. Достойные эмуляторы имеют большинство популярных консолей и компьютеров, однако в *Mednafen* нас заинтриговала поддержка относительно редких и малоизвестных систем, таких, как Neo Geo Pocket, WonderSwan и Virtual Boy.

Зависимости установить крайне легко: вам просто нужна SDL (и ее заголовки разработчика, при компиляции из исходника). *Mednafen* управляется из командной строки и не предлагает встроенного инструмента настройки, но в большинстве случаев вы можете просто запустить его с помощью файла ROM:

```
mednafen somegame.rom
```

Главный камень преткновения – это привязка к клавишам. В некоторых консолях можно долго молотить по клавиатуре, пока не найдешь нужную клавишу, если, конечно, не заглянуть в папку Documentation и файлы HTML для каждого устройства. В них указана настройки клавиш по умолчанию, а также настройки, изменяемые по вашему усмотрению для визуальных и звуковых элементов эмулятора. Очень радует, что клавиша с левыми кавычками ускоряет эмуляцию, прямо как в нашем старом фаворите, ZSNES.

Мы нашли пару GUI для *Mednafen*, но они только для Windows, и если вы любите эмуляторы и жаждете покодировать, вам есть куда приложить свои умения... LXF

› Большинство игр WonderSwan — только на японском языке, так что придется вам освежить свои знания иероглифов.

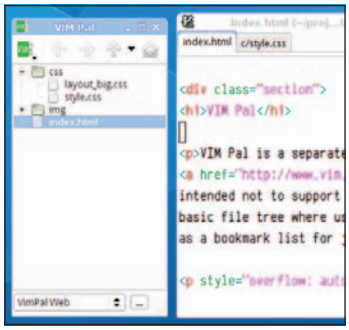
Также вышли

Новые и обновленные программы, тоже достойные внимания...

Glassomium 0.3
Забавный менеджер окон для сенсорных устройств.
www.glassomium.org

MoreAmp 0.1.28
Аудиоплейер, транскодер и программа записи CD, с мощным графическим эквалайзером.
http://sourceforge.net/projects/moreamp/

VIM Pal 1.2.0
Добавляет простой браузер файлов в текстовый редактор *Vim*.
http://vimpal.sourceforge.net



› Переходите на часто используемые файлы в *Vim* с этим удобным дополнением.

Salut à Toi 0.3.0
Клиент для XMPP/Jabber, с рабочим столом, консолью и множественными web-интерфейсами.
www.goffi.org

Siren Renamer 3.10
Инструмент для массового переименования файлов по имени, дате, расширению и т. д.
http://freecode.com/projects/siren-renamer

GnuAccounting 0.8.2
Написанное на Java кросс-платформенное приложение для бухгалтерских задач.
www.gnuaccounting.org

› Взаимодействие с ландшафтом — важная часть игрового процесса Zero-K.

На диске

Пробуйте новую операционную систему уже сегодня!

В странном мире календаря публикаций я обозреваю диски уже год. Это был год усердной работы, большого удовольствия и множества свободных программ. Я испробовал все основные и множество менее крупных дистрибутивов, получая от этого как хорошие, так и плохие впечатления. Я понял, что вселенная Linux намного больше, чем мне казалось раньше. Начиная со специализированных дистрибутивов (таких, как XMBCBuntu, Tails и Qubes), продолжая тем, что можно назвать дистрибутивами интерфейса (например, Zorin, SolusOS и CrunchBang), и заканчивая дистрибутивами-гигантами (в том числе Ubuntu, Fedora и OpenSUSE), здесь действительно есть все, что нужно. И если вы найдете время, вы сможете найти здесь все прелести ОС с открытым кодом.

Фермер призывает физика...

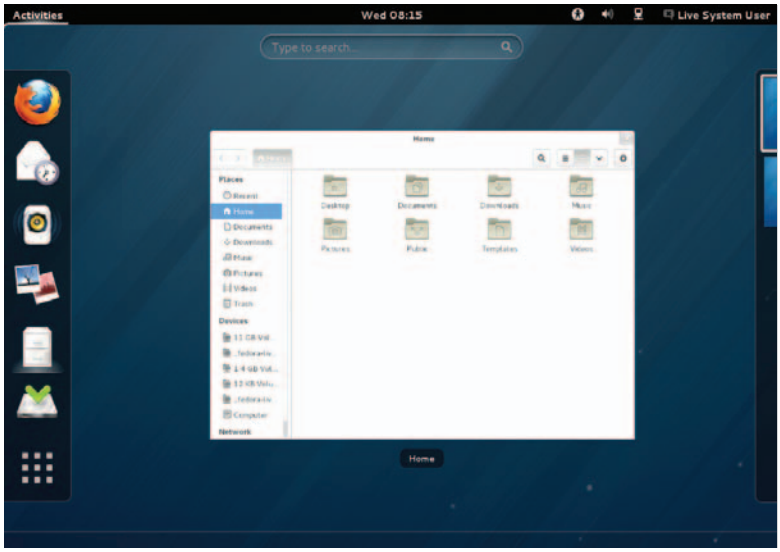
Fedora 18

Дистрибутив от сообщества Red Hat только что достиг нового номера версии. Fedora как-то задержалась, продвигая рабочий стол Gnome Shell и удумывая дурацкие имена. И в этой версии – Spherical Cow – ситуация не изменилась.

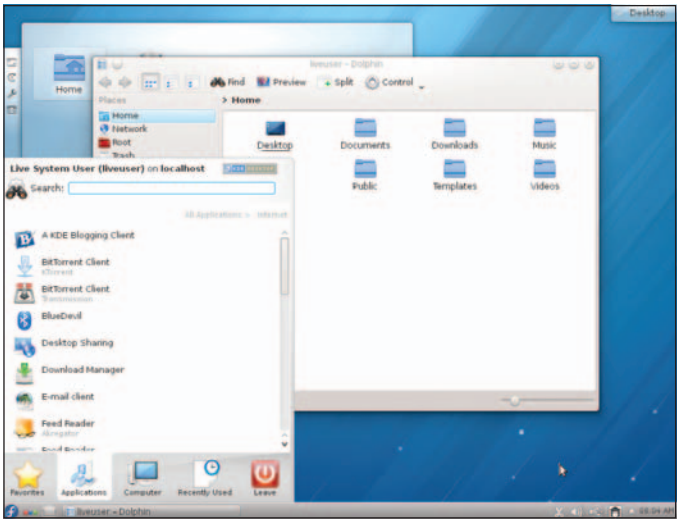
Мы в *Linux Format* любим Fedora, а вот к Gnome Shell испытываем смешанные чувства, поэтому мы создали собственное ответвление Fedora 18, которое идет с KDE и *Xfce*, а также с обычным Gnome. Решайте сами, чем вам пользоваться. После загрузки перед вами появится страница входа, где вы можете выбрать рабочий стол в Session Type, затем войти в систему под именем пользователя liveuser. Можете менять рабочие столы, выходя из системы и выбирая новый перед тем, как войти опять.

Вероятно, самую сильную критику в этом релизе вызывает обновленная программа установки *Anaconda*. Новая манера требует, чтобы пользователь скакал взад-вперед не вполне понятным способом. Немного пощелкав мышью, вы справитесь, но все же это довольно запутанно.

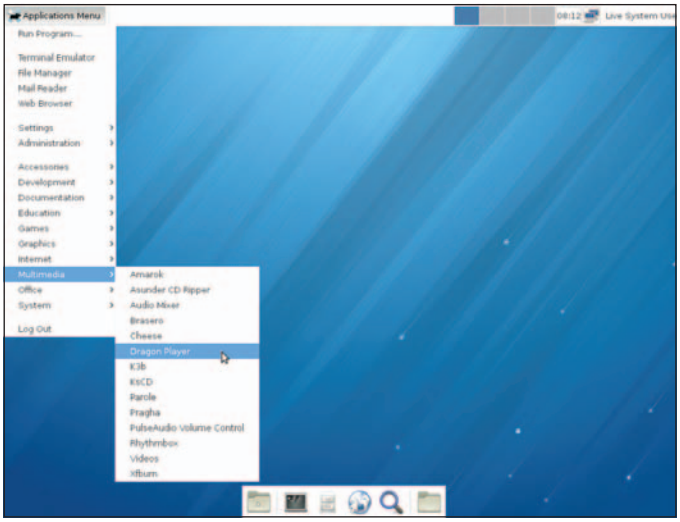
Но мы не хотим быть злопыхателями. В конце концов, не нравится вам работать в Gnome – ну и не работайте, а программу установки придется использовать только один раз. Fedora всегда была любимым дистрибутивом у тех, кто хотел быть в курсе всех последних достижений, что и остается по-прежнему в силе. Несмотря на некоторые недочеты, это все же отличный выбор. Для тех, кто использует Linux профессионально, Fedora – особенно привлекательная опция, поскольку ее внутренняя архитектура такая же, как в Red Hat Enterprise Linux, одной из самых популярных версий Linux для бизнеса.



› Дистрибутив сообщества Red Hat, Fedora, остается дистрибутивом Linux, который теснее всего связан с рабочим столом Gnome.



› KDE – рабочий стол для любителей все настраивать самим – предусмотрен как опция по умолчанию, и он загрузится, если вы не выберете ничего другого.



› Простота, удобство и легкость – вот отличительные черты *Xfce*, привлекающие к нему пользователей, которым не по душе ни KDE, ни Gnome.

Debian + OpenBox

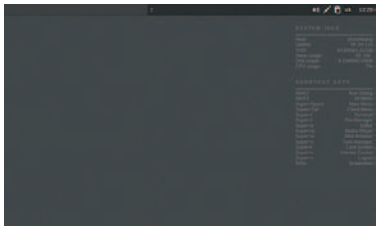
CrunchBang

Старый фаворит матерых линуксоидов, CrunchBang основан на *OpenBox* – менеджере окон, который специально создан в погоне за простотой. Не ждите от него изобилия графических прелестей, зато вам достанется простая, эффективная и мощная система, которая только и жаждет раскрыть свои возможности.

Как видно на экранном снимке, по умолчанию он неказист, но пусть это вас не пугает. Менеджер окон *OpenBox* можно настроить, чтобы он стал приглядней (если для вас это так важно), и он очень быстро работает почти на всех системах, на которых работает Linux.

CrunchBang основан на Debian, так что в менеджере окон вы обнаружите все ожидаемые программы (и даже больше, чем вам, вероятно, потребуется), которые только и ждут, чтобы вы их установили. Он также унаследовал знаменитую стабильность и свободу Debian, и вы можете смело использовать систему, не беспокоясь об этих проблемах.

Мы включили его в виде ISO-образа, так что вам придется перенести его на CD или USB-брелок, или запустить на виртуальной машине.



› CrunchBang: идеален для тех, кто любит серый цвет.

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.

Для учебы между кружками пива

UberStudent

Мы всегда охотимся за дистрибутивами, которые так или иначе выделяются из общей массы, и UberStudent привлек наше внимание. Как можно догадаться по названию, это в первую очередь дистрибутив для студентов, но это вовсе не означает, что он интересен только студентам.

Это отличная среда для тех, кто живет в режиме онлайн. В прошлом месяце мы рассказывали о Netrunner, практикующем такой же подход, используя рабочий стол KDE. В этом месяце мы включили UberStudent, который собран на *LXDE*. Выбор программ по умолчанию (большинство из них – web-приложения) тщательно продуман, чтобы вы получили все нужные функции, заполняя пробелы в web-приложениях оффлайн-приложениями – там, где это уместно.

Хотя он и нацелен на студентов, его могут счесть весьма удобным и другие. Например, в меню Education [Образование] есть следующие подменю: Books [Книги], Presentations [Презентации], Research and Writing [Исследовательская работа], Resources [Ресурсы], Self-Management [Самодисциплина], Student Stuff [Специально для студентов], Study Aids [Учебные пособия] и Subjects [Предметы]. Большинству покажутся полезными немало из этих подменю, и они относятся к областям, которым не уделяется внимание в большинстве дистрибутивов. Стоит загрузить его, хотя бы для знакомства с web-приложениями – вдруг какие-нибудь вам пригодятся.

› UberStudent помог нам открыть для себя Open Library, свободный сайт, где можно почитать книги онлайн.

Во время загрузки в режиме live появляется несколько ошибок в связи со сложностями мультизагрузки, но вы можете не обращать на них внимания. LXF

На диске

Помимо четырех симпатичных дистрибутивов, мы включили в состав диска весь код из руководств, все программы из Hotpicks и все программы, упомянутые в журнале, рядом с которыми вы видите значок **На диске**. Если вы вставите DVD в привод, 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.

Чтобы познакомиться с дистрибутивами, установите диск в дисковод, затем перезагрузите компьютер. После этого загрузится экран, где вы сможете выбрать то, что вам нужно. Если ваш компьютер загружается в обычном режиме, а не с диска, вам придется изменить настройки BIOS на загрузку с DVD. Для тех, кому захочется перенести содержимое диска на устройство USB, мы включили удобный скрипт. Чтобы узнать, как им пользоваться, загляните в **dvd2usb.html** на диске. Цифровые подписчики при желании могут применить **dd** для перемещения загруженного ISO на устройство USB:

```
sudo dd if=lxf169.iso of=/dev/sdX
```

при необходимости включив путь к загруженному ISO, где X изменяется согласно вашему устройству USB. **Это уничтожит все данные на диске. Если вы установите не тот диск, он может стереть все данные с вашего жесткого диска.**

LINUX FORMAT
Главное в мире Linux

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:

250 руб.

LXF165/166
Январь 2013

- **Raspberry Pi** Удумываем хитрые применения
- **Виртуализация** Как сделать из одного компьютера много
- **Byzantium** Узлы сети выходят из строя, а сеть работает
- **Патенты** Да, они – зло, и вот что каждому необходимо о них знать

LXFDVD: Linux Mint 14, Slackware 14 и еще 3 дистрибутива, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_165-166/
PDF-версия: shop.linuxformat.ru/elxf_165-166/

250 руб.

LXF167
Февраль 2013

- **Mint** Новый фаворит на скачках дистрибутивов
- **Офисные комплекты** Вооружение для планктона и не только
- **BTRFS** Очередная файловая система будущего
- **Взлом web-приложений** Не корысти ради, а в научных целях

LXFDVD: Linux Mint 14, Fedora 18 и еще 2 дистрибутива, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_167/
PDF-версия: shop.linuxformat.ru/elxf_167/

250 руб.

LXF168
Март 2013

- **Linux vs Windows 8** Победителя не придется угадывать до трех раз
- **Фоторедакторы** Пририсуйте друзьям усы и/или соберите фотки в альбом
- **System on a Chip** Целый компьютер на одном кристалле
- **Воскрешение из мертвых** Старый друг на новом железе

LXFDVD: Fuduntu, Netrunner, GhostBSD и **Mint аж в четырех вариациях**, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_168/
PDF-версия: shop.linuxformat.ru/elxf_168/

Ну, а если вы хотите быть уверенными, что не пропустите ни один номер журнала — оформите подписку! Помните, что все подписавшиеся на печатную версию журнала через www.linuxcenter.ru или shop.linuxformat.ru получают электронную версию в подарок!

Подписывайтесь на журнал на www.linuxformat.ru/subscribe/

Телефоны отдела подписки: Санкт-Петербург (812) 309-06-86, Москва (499) 271-49-54

Специальное предложение

Купите подборку журнала!

К нам в редакцию периодически приходят письма с вопросами, где можно купить предыдущие выпуски **LXF**. Если вы тоже этим озадачены, то в интернет-магазине «ГНУ/Линуксцентра» продолжается продажа журналов за 2011 год. Вы можете приобрести как отдельные номера изданий, так и подписки на 6 или 12 месяцев. **Спешите — журналов осталось не так уж много!**

shop.linuxformat.ru

Информация о диске

Внимательно прочтите это перед тем, как использовать LXFDVD!

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

Содержание

Сторона 1

ДИСТРИБУТИВЫ

CrunchBang 11-20130119 Дистрибутив на базе Debian с менеджером окон OpenBox, 32-разрядная сборка (ISO-образ)
Fedora 18 Live LXF Spin с рабочими столами Gnome, KDE и Xfce, 32-разрядная сборка (загрузка с **LXFDVD**).
UberStudent 3.2.0 Дистрибутив на базе Ubuntu с рабочим столом LXDE и набором приложений для студентов, 32-разрядная сборка (загрузка с **LXFDVD**).
5 дистрибутивов для Raspberry Pi Android 4, Arch Linux 2012-09, Plan 9, Raspbian Wheezy 2012-12, RiscOS 2012-11

Сторона 2

ДИСТРИБУТИВЫ

openSUSE 12.3 Установочный DVD, 64-разрядная сборка (загрузка с **LXFDVD**).

ДОКУМЕНТАЦИЯ: 10 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 Раймонда [Eric S. Raymond] «Собор и базар»
The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian
Dive Into Python Учебник по программированию на Python
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 – Грегом Кроа-Хартманом [Greg Kroah-Hartman]
System Administrators Guide Руководство по базовому администрированию Linux
GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

HOTPICKS

Avogadro 1.1.0 Редактор молекул
CPPCheck 1.58 Анализатор исходного кода C/C++
Erebus Игра, RPG «наведи-и-щелкни» (исходный код)
FatRat 1.2.0 beta 2 Менеджер закачек с поддержкой HTTP, HTTPS, FTP и BitTorrent
GTKDialog 0.8.3 Средство для создания диалоговых окон GUI из скриптов
LiteIDE Графическая среда разработки на языке Go
Mednafen 0.9.28 Эмулятор различных игровых консолей (Atari, Neo Geo, GameBoy, Nintendo, Sega и других)
Parcellite 1.1.3 Менеджер буфера обмена
PokerTH 1.0 Игра, сетевой многопользовательский покер
Safe-rm 0.9 Скрипт для безопасного удаления файлов, заменяет команду **rm**
Xowa 0.2.1 Программа для просмотра Wikipedia оффлайн

Пожалуйста, перед использованием данного диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИИ? Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕФЕКТНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу disks@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства; однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска.
К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Тираж изготовлен ООО «Уральский электронный завод». 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПТР ВАФ 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?

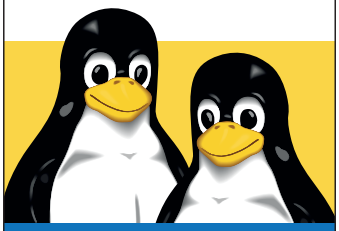
А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-нибудь с таким устройством? Вам не обязательно использовать Linux для записи дисков: подойдет любая операционная система, способная распознать пишущий привод (см. выше).

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.

ФОРУМ №1 ДЛЯ ВСЕХ ПОЛЬЗОВАТЕЛЕЙ LINUX

LINUXFORUM.RU

LinSoft.Info
Путеводитель по программному обеспечению для GNU/Linux



WWW.LINSOFT.INFO

Linux по-русски
Библиотека книг, статей и переводов о Linux
WWW.RUS-LINUX.NET

Progressive Operating Systems
CANONICAL partner agent
ubuntu
официальная поддержка
Ubuntu Advantage
для вашего бизнеса
думайте
действуйте
меняйтесь
+7 495 625-19-74
http://www.p-os.ru

LINUX FORMAT
Главное в мире Linux
Как разместить рекламу в разделе Classifieds?

1/1 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

Linux center
www.linuxcenter.ru
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

UnixEducation Center
Россия, 190000, Санкт-Петербург
Черноморский переулок, дом 4
Тел.: + 7 (812) 611-1575
ORACLE Symantec. redhat
PEARSON VUE AUTHORIZED TEST CENTER
PROMETRIC TEST CENTER
RHCSA / RHCE Certification

Футболки GNU/Linux
БОЛЕЕ 100 ВИДОВ. ВСЕ РАЗМЕРЫ И ЦВЕТА

WWW.LINUXCENTER.RU/SHOP/GIFTS/

ВЗЛОМ СЕТИ

В майском номере

Взлом Сети

Есть на свете злые люди, которые только и мечтают до вас добраться. Мы рассмотрим ряд самых популярных атак из Сети и обучим вас самозащите.

Samba 4

Ну да, да, мы знаем, что Linux лучше, но иногда приходится поработать и с Windows. И это стало проще.

Мощь дистрибутивов

Вы – крутой? Мышь презираете? Мы подобрали лучшие дистрибутивы для матерых линуксоидов.

XBMC

Мы обозрели его в этом номере, но через месяц копнем глубже и заберемся в потрясающего Frodo.

Содержание будущих выпусков может меняться – вдруг мы переберемся в Брюссель...

LINUX FORMAT
Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 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, Елена Толстякова, Ирина Шулакова
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 «ЛапинЪ»
188330, Ленинградская обл., Гатчинский р-н, пос. Сиверский, Вокзальная ул., 4
Заказ 7744

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 Эфраим Эрнандес-Мендоса [Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com
Выпускающий редактор Гэри Уокер [Gary Walker] gary.walker@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ
Джоно Бэкон [Jono Bacon], Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Бен Эверард [Ben Everard], Марко Фиоретти [Marco Fioretti], Эндрю Грегори [Andrew Gregory], Дэвид Хейвард [David Hayward], Грэм Моррисон [Graham Morrison], Джонатан Робертс [Jonathan Roberts], Майк Сондерс [Mike Saunders], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Евгений Балдин, Артем Зорин, Константин Калгин, Дмитрий Кузнецов, Андрей Пономаренко, Владимир Рубанов, Алексей Федорчук, Максим Черепанов

Художественные ассистенты Стейси Блэк [Stacey Black], Джон Мак-Аллистер [John McAllister], Мэтт Ортон [Matt Orton]
Иллюстрации Шейн Коллиндж [Shane Collinge], Elly Walton Illustrations, iStockPhoto, Саймон Миддлвик [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 неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на CD или DVD-диски, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет 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

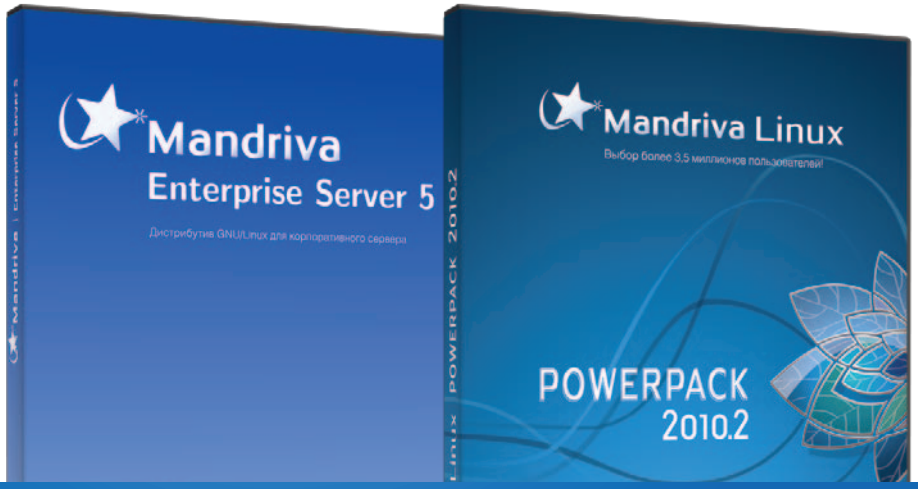
За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт http://www.futureplc.com

Future MEDIA WITH PASSION

© Linux Format 2005
© Future Publishing Ltd 2005

16+

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO



Mandriva
www.mandriva.ru

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

MANDRIVA УЖЕ ИСПОЛЬЗУЮТ: 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.

С вопросами по продуктам Mandriva обращайтесь в **ГНУ/Линуксцентр**!

Москва +7 (499) 271-49-54
Санкт-Петербург +7 (812) 309-06-86

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

Linux center

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.

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