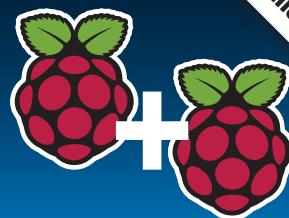


LINUX FORMAT

Главное в мире Linux



Мощь команды Pi на Python

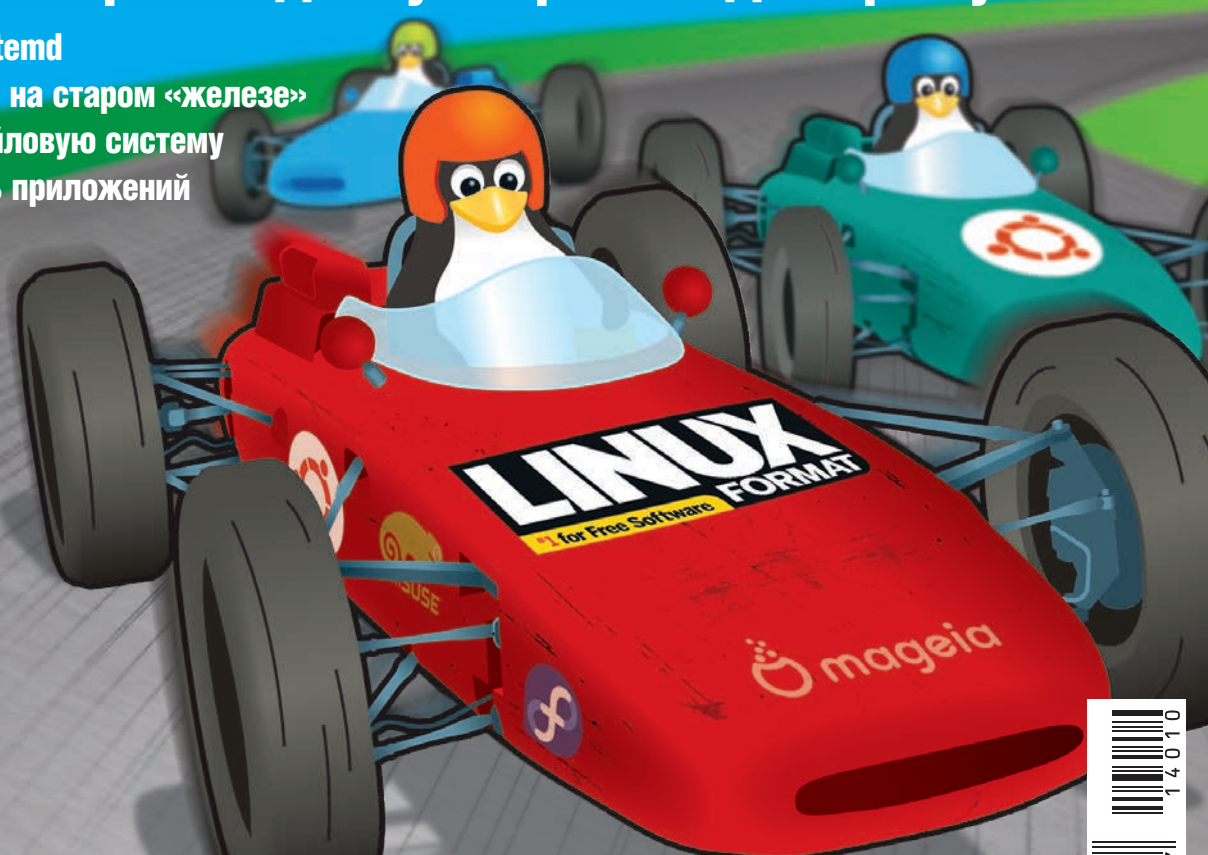
» Распределенные вычисления — это просто, если знать, как!

Октябрь 2014 № 10 (188)

РАЗГОН LINUX

Хитроумные трюки для ускорения дистрибутива

- » Изощряйтесь с Systemd
- » Экономьте ресурсы на старом «железе»
- » Оптимизируйте файловую систему
- » Увеличьте скорость приложений



ПЛЮС!
Не так страшно ядро Linux, как его малюют
с. 76



Открытое образование

« Возврат информатики в школы — это здорово, и я горжусь, что имею к этому отношение »

Клайв Бил Linux шагнул в соседнюю школу с. 40

Также в номере...

Распознавание речи

» Как управлять Linux командирским голосом с. 24



Web-серверы
Внутри Nginx
» Сервер нового поколения

Хостинг фото
Сервер Lychee
» Управляем фотографиями через Сеть

Гид по дистрибутиву
Arch Linux
» Стильный и мощный, с плавающими релизами

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

Linux center
www.linuxcenter.ru



iTeleRadio

ИНТЕРАКТИВНОЕ РАДИО&ТЕЛЕВИДЕНИЕ

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

Мы ускоряем наши компьютеры. И мы спросили наших авторов: как бы они перделали свой ПК в суперкомпьютер?



Джонни Бидвелл

У меня еще на ходу один из первых Eee PC, и я им горжусь. По времени загрузки/входа (в премилый рабочий стол Fluxbox) это самая быстрая машинка из всех, что я имел. Чудеса творят твердотельный SSD-диск и ОС, урезанная до голы необходимости. Еще один хитрый трюк — засунуть свой профиль Firefox в ОЗУ.



Дэвид Хейвард

Кроме вложения в это дело больших средств, чем позволила бы миссис Хейвард, я бы загрузил цифровую копию Джонни Делпа из «Превосходства» и вежливо попросил бы его придать мне монстрособности. Правда, при моей везучести я скорее нарвусь на какого-нибудь психического Джеффа Файи.



Эндрю Моллетт

На моем компьютере стоит Ubuntu Trusty Tahr, и до суперского статуса ему недостает только кофейного порта USB, чтобы он мог по запросу вливать в меня животворную «Арабику». А CD-привод уже давненько приобрел кольцевидную форму — самое оно для изготовления лучших в мире пышек с повидлом.



Лес Паундер

Для ускорения Linux я присоветовал бы процессор на нейронной сети, сверхпроводящей при комнатной температуре (естественно). Неплохо также рассмотреть возможность интеграции идеи в серию двуногих, гусеничных или летающих роботов. Голубое небо только через сетевой проект и увидишь.



Маянк Шарма

Фи, какой примитивный вопрос. Я бы раскопегарил свой верный телепортатор, нацепил кожаное пальто до пят и отправился в 1970-е со своим четырехъядерным ноутбуком. Адмирал Хупер точно отпал бы при виде такого продвинутого девайса. Может, даже назвал бы линкор моим именем...



Валентин Синицын

Судя по тому, что я недавно видел в кино, нужно побольше лампочек и непременно черно-зеленый винтажный монитор. Такой компьютер подбирает пароль к сети Пентагона за несколько секунд. И, разумеется, надо самостоятельно пересобрать ядро Linux.



Архитектура и излишества

» Проблема «разгона» системы столь же стара, как и сама индустрия ИТ. Решить ее можно двумя способами: «кошельком» (купить более производительное «железо») или «интеллектом» (настроить систему надлежащим образом). В последнем случае главная рекомендация проста — убрать все лишнее. Там, где производительность по-настоящему критична, именно возможность убрать «архитектурные излишества» становится важнейшим критерием выбора ОС. Для доказательства достаточно взглянуть на Top-500 суперкомпьютеров (<http://www.top500.org/lists/2014/06/>).

У большинства проприетарных систем возможности «отсечения лишнего» весьма ограничены. Решение архитектора непрекрасно и обсуждению не подлежит. А если кто не согласен... ну, он просто ничего не понимает в высоком искусстве...

В мире свободного ПО все наоборот: пользователь может выбрать дистрибутив с наиболее подходящими умолчаниями, а потом оптимизировать настройки и заменять компоненты, вплоть до полной пересборки всего кода. Здесь каждый сам себе архитектор.

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

Общие вопросы: info@linuxformat.ru

Проблемы с дисками: disks@linuxformat.ru

Вопросы распространения: sales@linuxformat.ru

Сайт: www.linuxformat.ru, группа «ВКонтакте»: vk.com/linuxform

» Адрес редакции: Россия, Санкт-Петербург, пр. Медиков, 5, корп. 7

» Телефон редакции: (812) 309-0686. Дополнительная информация на с. 112

V Международная
конференция



MANAGED SERVICES

30 октября

EVENT-ХОЛЛ «ИнфоПространство»
Москва, 1-й Зачатьевский пер., д. 4

2014

Оптимизация затрат операторов, аутсорсинг сетевой и ИТ-инфраструктуры, совместная эксплуатация сетей связи

Основные темы:

- Аутсорсинг телекоммуникационной инфраструктуры в России и в мире. Опыт крупнейших проектов отрасли
- Оценка рисков оператора при передаче сетей на обслуживание сторонним организациям
- Инициативы регулятора по легализации совместного использования инфраструктуры различными операторами
- Предварительные итоги реализации операторами сотовой связи программы по покрытию магистральных автотрасс услугами ПРТС
- Развертывание сетей связи нового поколения с использованием управляемых услуг
- Практика строительства волоконно-оптических линий связи на воздушных линиях электропередачи для сотовых операторов
- Практика ИТ-аутсорсинга, управления ИТ-инфраструктурой и ИТ-процессами в интересах операторов связи
- Совместное использование активной инфраструктуры (Active Network Sharing). Мировой опыт и перспективы преодоления нормативно-правовых барьеров в России
- Бизнес-модель организации сети инфраструктурных площадок совместного использования для операторов сотовой связи

Докладчики:



Руслан Вагизов,
исполнительный директор,
ICL Services



Владимир Валькович,
руководитель
департамента
технического развития
и эксплуатации сети,
Orange Business Services



Сергей Македонский,
президент, Ассоциация
стратегического
аутсорсинга «АСТРА»,
генеральный директор,
in4media/Forrester Research



Юрий Маленков,
директор
по планированию
и развитию сетей
доступа,
ОАО «ВымпелКом»



Юрий Панченко,
заместитель директора
сервисного центра
по организации
производства,
ЗАО «Инфосистемы Джет»



Михаил Романов,
партнер, руководитель
практики телекоммуникаций,
медиа и высоких технологий,
Ernst & Young (CIS)



Юрий Рубцов,
начальник департамента
по управлению качеством
функционирования сети
и эксплуатационными
затратами, ОАО «МегаФон»



Виталий Слизень,
член совета директоров,
Inoventica



Валерий Тихвинский,
заместитель генерального
директора по инновационным
технологиям,
ООО «АйКомИнвест»



Андрей Яценко,
генеральный директор,
ООО «ГорСвязьСервис»

Организатор:



Серебряный спонсор:



Бронзовый спонсор:



Эксперт-партнер:



При участии:



При поддержке:



Официальный
аналитический
партнер:



Для регистрации: +7 495 933 5483, conf@comnews.ru, www.comnews-conferences.ru/ms2014

Содержание

Быстрее, чем летит ласточка без груза (так говорят в Африке).

Обзоры

Red Hat Enterprise 7 14

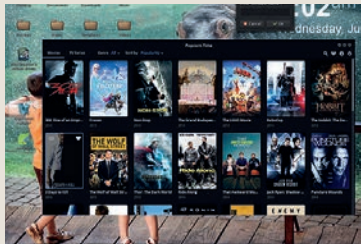
Не подобрать ли вам новую Шляпку — 64-битную? Прибывает RHEL 7, да не простой, а с поддержкой контейнеров и автоматических обновлений без переустановки.

КаОS 2014.06 16

Независимый дистрибутив — из тех, что надо смаковать, а не заглатывать залпом, как водку на буйном мальчишнике.

MakuluLinux KDE 6.0 17

Дистрибутив, не похожий на других, выбрал себе честолюбивую цель «предоставлять целый мир возможностей».



➤ Возвращение “Big Momma” с чемоданами пакетов и пост-установочных скриптов.

Opera Developer 24 18

Толстые тетки, запевай! Opera вновь выходит на сцену — и выкатывает для Linux свежайшие функции, включая Discover, Stash, Speed Dial и Opera Turbo.

HP 255 G1 19

Малобюджетный ноутбук с предустановленным Ubuntu, от сами знаете каких больших парней.



➤ Наклейка-логотип Ubuntu мирно соседствует на корпусе с клавишей Windows.

Raspberry Pi Model B+ 20

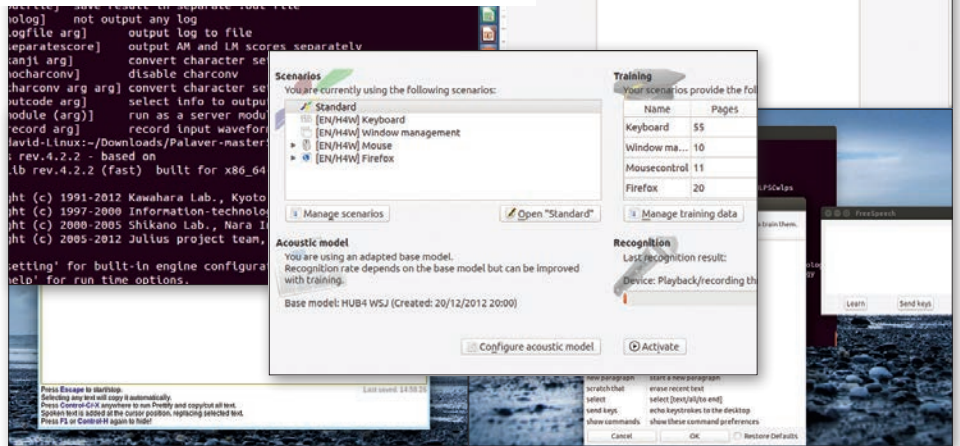
Команда B получила целую кучу дополнительных расширений, переключив хватательный рефлекс на себя: руки в первую очередь тянутся к ним.



Вдарим по газам своего ПК, уменьшим время загрузки и извлечем из системы максимум с. 30

Разгон Linux

Сравнение: Распознавание речи с. 24



Люди говорят

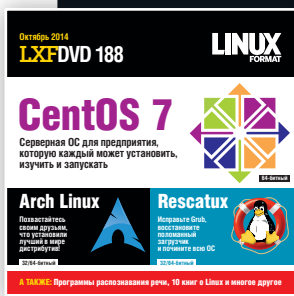


“ Возврат информатики в учебные планы откроет огромные перспективы ”

Клайв Бил о философии Open Source в образовании с. 40

На вашем бесплатном DVD

Побалуйте себя и любимых подпиской на LXF!



CentOS 7

» Серверную ОС для предприятия — всем

Rescatux 0.32

» Проблемы как лапой... в смысле, рукой снимает

ПЛЮС: HotPicks и коды учебников **с. 106**



Доступно в AppStore!

www.linuxformat.ru/subscribe

Ищите в этом номере

Открытое образование 38

Мы не нуждаемся в контроле за мыслями.

4K: уже играет 42

Знакомьтесь: рабочая среда будущего.

Arch Linux 46

Дружите с ним — и действуйте функционально.



Академия кодига

Распределенные вычисления 84

Застрянем в колючих кустах: от Малины к Ежевике. Построим кластер из Raspberry Pi (он же Bramble), и пускай Python грызет код.

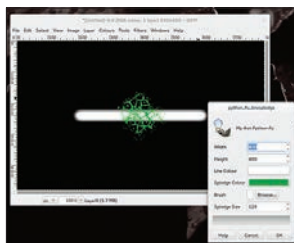
Напишем плагин GIMP ... 88

Расширим популярную открытую программу обработки изображений,

добавив при помощи Python приятный эффект боке.

Старый конь 92

Язык-ветеран Fortran не сдает позиций благодаря мощному научно-техническому лобби. Численные расчеты — его козырь, и в свободном обращении ходит множество готовых решений.



Постоянные рубрики

Новости 6

«РОСА» братается с «Рикором», а атомщики — с железнодорожниками; вышла куча новых релизов, КНР порывает с Microsoft, а Центризбирком всерьез намеревается переходить на СПО.

Новости Android 22

Фрагментация неоднозначна по последствиям, на подходе — 64-битный процессор для телефонов, навигатор все хорошеет, юный Суанопен отбивает поклонников у Google, а Lenovo затевает выпуск планшета.

Сравнение 24

Поговори-ка ты со мной... а вот и не гитара, а компьютер. Тестируем системы распознавания речи для Linux. *FreeSpeech, Julius, Palaver, Simon, Speech-App.*

Интервью LXF 38

Марк Чемберс, Клайв Бил и Раджай Наик приходят в восторг от перспектив, открываемых Linux и Raspberry Pi в обучении информатике в школах (и не только). За этим — будущее нации.

Рубрика сисадмина 50

Доктор признается, что в жизни не все получается гладко, и рассматривает интеграцию систем на Linux с сетями на Windows, которые используют Active Directory.

Ответы 96

ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!
Нейл Ботвик — про установку редактора *uED* в Wheezy; *Grub* и MS-DOS; воспроизведение видео в Chrome; воспроизведение DVD; незагружаемые DVD; беспроводное подключение в Xubuntu.

HotPicks 100

Отдайте горяченького! Лучшие в мире новинки свободного ПО: *CodeWorld, Kangas Sound Qi, LFTP, Mailpile, n2048, Perl 5.20, Rescatux, Seafile, Seagull Soup, TeXStudio, The Music Suite.*

Диск Linux Format 106

Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108

И лишились жизненно важного учебника/статьи/обзора... Скорей обращайтесь к нам, пока старые номера еще не разошлись.

Через месяц 112

Решаем все проблемы Linux, ставим Linux на устройства с Android... и многое другое.

Учебники

Навыки в терминале Учетные записи 56

Создавайте пользователей, задавайте пароли и упивайтесь правами собственности.

Lychee Сделаем сервер фото 58

Делитесь фотографиями с родными и друзьями, создав собственный хостинг. Его потянет даже Raspberry Pi.

Systemd Локальные сети 62

Для несложной работы с локальной сетью приспособим *Systemd* и *Networkd*.

Nginx Настроим web-сервер 64

Повозимся с web-сервером нового поколения, который бросил вызов *Apache*.

OpenStack Swift и Heat 68

Храните объекты и упорядочивайте инфраструктуру своего облака.

PHP/SQLite Пишите письма 72

Напишем скрипты для почтовых рассылок с личного сервера, способных обдурить спам-фильтры.

Ядро Linux Заглянем внутрь 76

Модификация ядра Linux — отнюдь не удел одних только бородачатых гуру... Ядро — это всего лишь программа!

Электроника — умельцам ПЛИС 80

Благодаря невероятной гибкости программируемой логики можно собирать устройства любой сложности.



В ЭТОМ НОМЕРЕ: Наш ARM-сервер » Угрозы с облаков » Свежие релизы
» ОС для КНР » СПО выбирает » Союз атомщиков и РЖД

ARM В РОССИИ

Начало сотрудничества

«Рикор» и «РОСА» объединяют усилия.

Компания «Рикор.ИТ» создает хранилища данных и коммуникационное оборудование, а также проектирует и производит энергоэффективные серверы EcoServer — это высокоэкономичная масштабируемая платформа на базе процессора Marvell ARM Armada XP для ИТ-инфраструктуры современных ЦОД и облачных сервисов. «НТЦ ИТ РОСА» занимается разработкой ОС. Их сотрудничество позволит создать уникальный отечественный ARM-сервер на ОС от «РОСА», отвечающий интересам коммерческих структур и госорганов, в т.ч. ВПК, к энергоэффективным серверным вычислениям.

«ARM-архитектура — перспективное направление в развитии серверных платформ. Их также называют «экосерверами» благодаря низкому энергопотреблению,



» **Аркадий Тагиев,**
гендиректор «НТЦ
ИТ РОСА».

экономии площади за счет высокой плотности установки, сниженным требованиям к охлаждению и т.д. Создание совместного продукта с компанией «Рикор.ИТ» позволит выйти на качественно новый уровень комплексной разработки программно-аппаратных комплексов», отметил гендиректор «РОСА» Аркадий Тагиев.



» Рубрику готовил
**АНДРЕЙ
ГОНДАРЕНКОВ**

«Это позволит российским ИТ-компаниям не зависеть от импортных производителей как аппаратного серверного оборудования, так и программной составляющей. Мы очень заинтересованы в законченном решении, которое совмещало бы как наши инновации в области разработки серверного оборудования, так и инновации наших партнёров, которые разрабатывают операционные системы на базе GNU/Linux. В сотрудничестве с «НТЦ ИТ РОСА» мы видим огромный потенциал», комментирует заместитель директора по развитию «Рикор.ИТ» Олег Кузнецов.

В свете набирающего популярность направления импортозамещения, на архитектуре ARM в России планируется также развивать производство чипсетов и т.д.

НЕМНОГО ФУТУРОЛОГИИ

Прогноз от Gartner

Мобильность, социализация и облака несут угрозу.

Согласно отчету компании Gartner, специализирующейся на исследованиях и консалтинговых услугах для информационного рынка, общемировые расходы на обеспечение информационной безопасности (ИБ) в 2014 году превысят \$71 млрд, увеличившись, против прошлого, на 7,9%, а в сегменте предотвращения потери данных — на рекордные 18,9%. В 2015 году это будет уже \$76,9 млрд. Стимул к росту расходов — стремительное расширение применения мобильных и «облачных» сервисов, а также соцсетей. Организации яснее

осознают многочисленные угрозы, сопровождающие развитие этих технологий.

Директор Gartner по исследованиям Лоуренс Пингри [Lawrence Pingree] отметил «демократизацию угроз». Вредоносное ПО для «продвинутых» атак злоумышленникам сейчас легко приобрести онлайн. Это осложнило жизнь отделам по безопасности, но и привело к пониманию, что ИБ нельзя рассматривать как обычную ИТ-функцию или статью расходов в бюджете предприятия. Ресурсы стали перенаправляться с администрирования средств обеспечения ИБ на организацию

системного процесса управления инцидентами и минимизации их последствий.

Рост популярности хостинга приложений и развитие инфраструктуры также чреваты реформами в секторе обеспечения ИБ. Увы, многим организациям не хватает персонала с должными навыками. Чтобы не отставать от хакеров, более половины компаний к 2018 году обратятся к услугам специализированных фирм по защите данных, управлению рисками и безопасности инфраструктуры, прогнозируют аналитики Gartner. Они уверены, что в 2015 году около 10% от всех возможных мер защиты предприятий будут реализованы через облачные сервисы, в особенности в малом и среднем бизнесе.

На борьбу с киберугрозами нацелены законы о конфиденциальности и защите данных, находящиеся на разных стадиях реализации или планируемые к принятию в Австралии, ЕС, Сингапуре и Малайзии.

РЕЛИЗЫ

KDE SC 4.14

Вышел крупный релиз KDE Software Compilation.

Сообщество KDE представило очередной релиз своего популярного настольного окружения, посвятив его памяти одного из многолетних участников проекта, Фолькера Ланца [Volker Lanz]. Развитие KDE как подборки ПО (Software Compilation) завершено. Поскольку сейчас основные усилия разработчиков Plasma Workspaces и KDE Development Platform направлены на отладку задействованных в пользовательском окружении Plasma 5 механизмов взаимодействия с пользователем и перевод программ на набор библиотек KDE Frameworks 5, версия KDE Software Compilation 4.14 содержит только обновление приложений и исправление ошибок.

Так, в мультипротокольном клиенте мгновенного обмена сообщениями *Kopete* реализована поддержка прокси SOCKS5 для работы по протоколу ICQ (раньше было можно применять только HTTP); для учетных записей Jabber по умолчанию включена поддержка голосовых вызовов (и для протокола Google libjingle, и для обычного jingle); обновлена версия

библиотеки *libiris*, реализующей в *Kopete* протокол jabber/xmpp.

Cantor, интерфейс для интегрированных в KDE математических приложений, получил движок Lua; для данных в LaTeX поддерживается кодировка UTF-8; у движка Python 2 появилась поддержка расширений для линейной алгебры и рисования графиков; поддержка скриптов *autorun* добавлена в движки Python 2, *Scilab*, *Octave*, *Sage*, *Maxima*, *Qalculate* и *KAlgebra*.

Игра *Kanagram* получила новый интерфейс на базе QML.

Выполнен внутренний рефакторинг кода *Okular*, программы для просмотра документов.



➤ Переливы синего и лилового в Plasma выглядят стильно.

В текстовый редактор *Kate* включены новые правила подсветки синтаксиса языков программирования; усовершенствован vi-режим; автоматически сворачиваемые блоки комментариев высвобождают экранное пространство; улучшена поддержка и автоматическая генерация системных цветовых схем.

К компонентам среды UML-моделирования *Umbrello* добавлены порты UML2; улучшен импорт элементов C++; реализована новая функция «поиск в диаграмме»; поддерживается загрузка файлов (без диаграмм) *ArgoUML* и *Embarcadero*; усовершенствована загрузка *Rose*-моделей (добавлена поддержка контролируемых узлов и диаграмм классов).

Помимо исправления ошибок, в файловом менеджере *Dolphin* выполнен ряд мелких улучшений (подсветка текущего элемента панели Places выглядит симпатичнее; во всех окнах уведомление о свободном пространстве в строке состояния поддерживается в актуальном виде; начат процесс рефакторинга кода класса, отвечающего за отображение главного окна *Dolphin*).

Для виртуального глобуса *Marble* расширен список поддерживаемых функций KML; файловый менеджер *Dolphin* отображает миниатюры *.gpx*, *.kml* и других типов файлов, поддерживаемых *Marble*; новая политическая векторная карта отображает государственные границы различными цветами.

ДИСТРИБУТИВЫ

Обновление Qt Creator 3.2.0

IDE получила добрые две сотни изменений.

Вышла обновлённая версия *Qt Creator* — свободной кроссплатформенной интегрированной среды разработки приложений (IDE) с использованием инструментария *Qt*. В число языков, поддерживаемых этой IDE, входят как классические C и C++, так и основанный на JavaScript декларативный язык QML.

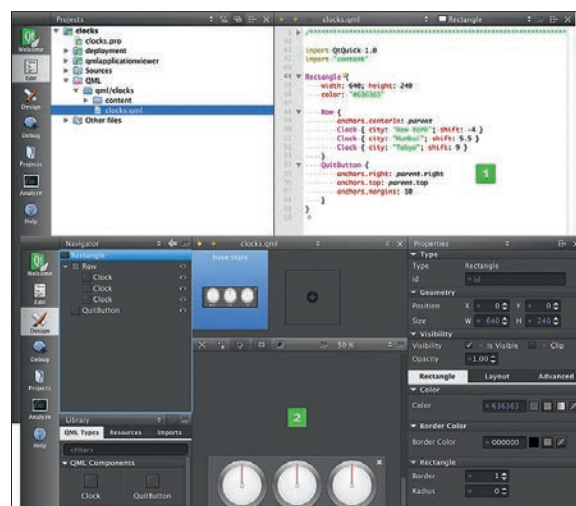
Qt Creator 3.2.0 содержит около 200 изменений, принятых от 50 разработчиков. Среди наиболее интересных новинок:

- возможность редактирования текста во всех столбцах выделенного блока: так, можно редактировать все выбранные строки;
- в модель кода C++ добавлена поддержка реализованных в стандарте C99 назначенных инициализаторов и сцепленных

строк, внесены улучшения в обработку кодировок и поддержку лямбда-выражений;

- предусмотрена возможность вывода контекстной справки в отдельном окне;
- поиск, запускаемый комбинацией клавиш Ctrl+F, ведётся в дополнительных панелях (например, в панели дерева проекта);
- улучшены производительность и стабильность *QML profiler*.

Пользователи Enterprise *Qt Creator* получили возможность применять *QML profiler* для отладки использования памяти JavaScript в QML. А вот поддержка Mac OS X 10.6 (Snow Leopard) в *Qt Creator 3.2.0* прекращена, из-за отсутствия поддержки в этой ОС от Apple некоторых элементов протокола C++11.



➤ Окно *Qt Creator* напоминает приборную панель космолета.

ЭТО ОБЛАКА

SUSE Cloud 4

Обновлена версия платформы с открытым исходным кодом.

Одной из главных новинок предназначенного для создания частных облачных инфраструктур IaaS (Infrastructure-as-a-Service) продукта SUSE Cloud 4 является поддержка расширенных возможностей платформы VMware в пределах комплекса OpenStack. Кроме того, в новой версии облачной платформы от SUSE реализована полная поддержка распределённой файловой системы Ceph.

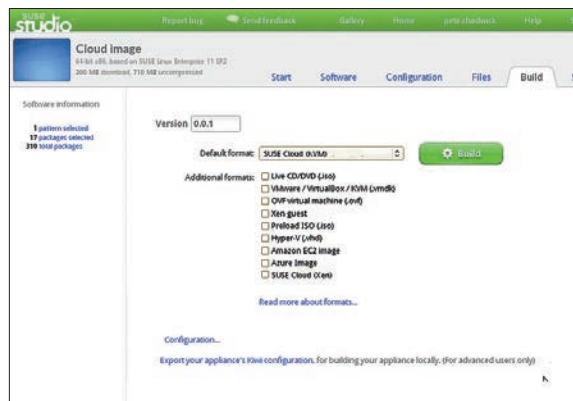
Решение SUSE Cloud 4, обеспечивающее возможность развернуть на собственных серверах организации локальную облачную инфраструктуру, основано на последней версии OpenStack (Icehouse) и предоставляет корпоративным клиентам следующие преимущества:

» снижение организационных затрат на создание решений для программно определяемого хранилища за счёт применения распределённой файловой системы Ceph (процедура установки позволяет

автоматически настроить и развернуть кластеры Ceph);

» расширение интеграции между окружениями VMware vSphere и OpenStack. В дополнение к реализованной в прежних версиях поддержке узлов VMware vSphere, решения для виртуализации сетей VMware NSX, драйверов для блочных хранилищ vSphere, в SUSE Cloud 4 добавлена поддержка VMware Virtual SAN — технологии виртуализации ресурсов хранения с прямым подключением, в которой отдельные накопители объединены в общее распределённое хранилище данных с высокой производительностью и отказоустойчивостью;

» упрощённый процесс развёртывания нагрузки повышает уровень масштабируемости и автоматизации. В составе SUSE Cloud поставляются база данных, балансировщик нагрузки и система ограничения доступа как сервис (firewall-as-a-service).



» Продвинутым пользователям есть с чем познакомиться.

В состав SUSE Cloud 4 входит SUSE Studio, средство интеграции с конструктором образов виртуальных окружений, а также SUSE Manager, решение для управления дистрибутивами в физической, виртуальной и облачной среде. На сайте производителя доступна для загрузки 60-дневная пробная версия.

НАЦИОНАЛЬНОЕ ПО

ОС для Поднебесной

Китай намерен избавиться от зависимости.

Китай близок к ликвидации зависимости от программных продуктов Microsoft Corp, Google Inc и Apple Inc: как сообщает агентство Синьхуа, в октябре в этой стране будет завершено создание ОС собственной разработки. В течение одного-двух лет национальная ОС в КНР заменит зарубежные на настольных компьютерах, а в течение трех-пяти лет — и на мобильных устройствах. Сотрудник Инженерной академии КНР Ни Гуаньнань [Ni Guangnan], руководитель созданного в марте Альянса по разработке ОС, рассказал, что национальная ОС будет поддерживать магазины приложений. Основными причинами создания собственной ОС названы прекращение с апреля 2014 года поддержки Windows XP и официальный запрет на использование Windows 8 на компьютерах госучреждений КНР. Эти события предоставили китайским разработчикам отличные шансы для продвижения собственной системы, однако отрасль

сильно нуждается в дальнейшем развитии и инвестициях. «Создание условий, позволяющих нам бороться с Google, Apple и Microsoft — это и есть ключ к успеху», уверен Ни Гуаньнань.

В марте прошлого года Китай заявил, что компания Google, при помощи своей мобильной ОС Android и дискриминации некоторых местных фирм, получила слишком широкий контроль над производством смартфонов в КНР. В течение года неуклонно накалялась напряжённость в сфере кибербезопасности между КНР и США, возникшая после заявления Эдварда Сноудена о «лазейках [backdoor]», внедряемых в производимые в США аппаратные средства. В свою очередь, в мае министерство юстиции США предъявило обвинения пяти гражданам Китая по нескольким эпизодам промышленного шпионажа.

Китай и ранее вёл работы по созданию собственных ОС. Так, в 2000 году вышла ОС Red Flag Linux, разработка которой

частично была финансирована Министерством информации. В том же году Red Flag Linux рекомендовали как замену Windows 2000 на всех компьютерах в госучреждениях КНР, поскольку в это время между правительством Китая и Microsoft возникли трения (чему немало способствовало широкое распространение в КНР пиратских версий ПО от Microsoft). Тем не менее, Красный Флаг (Red Flag) так и не взвился, и компания, осуществлявшая его разработку, была закрыта. Теперь же Красный Флаг — ОС, а не компания — возрождается.

В номере People's Post and Telecommunications News от 20 августа сообщалось о приобретении активов Red Flag Software за 38,62 млн юаней (\$6,3 млн) промышленной группой Penta Wan Jing Information Technology Industry Group. Ни Гуаньнань одобрил эту сделку: по его мнению, обновленная Red Flag будет способствовать планам по созданию отечественной ОС. Название новой ОС пока не огласили.

softline®

Cloud Software Hardware Services

20+

Years in IT

IT-архитектура вашего бизнеса



корпоративные
мобильные решения
на платформе Microsoft

SaaS

CLOUD

PROJECTS

гибридные облака

SOLUTIONS

remote workplace

SAM

поставка аппаратных
решений
юридическая
поддержка

education

Mirapolis

SECURITY
учебный
центр

разработка IT-стратегий

hardware
ERP&CRM
CAD
CAM
CAE

VIRTUALIZATION

web-разработка legal support

защита
персональных данных

SOFTWARE

публичные облака

САПР и ГИС

LINUX

программное
обеспечение

BI & BigData

SERVICES

WinSpace

информационная
безопасность

решения Microsoft

data leak
prevention

PaaS

разработка информационных систем

корпоративный портал
DeskWork

open
source

техническая поддержка

search

flexible

CONSULTING

PLATFORM

IaaS

частные
облака

protection

hosting

technical support

16+

8 (800) 100-00-23

+7 (495) 232-00-23

www.softline.ru

info@softline.ru

СПО И ОБЩЕСТВО

Выбирает СПО

Центризбирком откажется от Windows в ГАС «Выборы».

Информационная кампания по импортозамещению охватила российскую выборную систему. На письменную просьбу Людмилы Бокковой, председателя временной комиссии Совета Федерации по развитию информационного общества, рассмотреть возможность отказа от ОС Microsoft Windows, на которой сейчас работает Государственная автоматизированная система (ГАС) «Выборы», руководитель Федерального центра информатизации при ЦИК Михаил Попов ответил: Центризбирком планирует разработку новой версии ГАС «Выборы», на базе отечественной ОС. Центризбирком разделяет озабоченность членов Совета Федерации проблемой импортозамещения ПО. Более того, проектирование новой ОС уже началось. Подготовлены экспериментальные образцы и готовится серийный выпуск АРМ избирательных участков, которые будут работать под управлением GNU/Linux. При этом, подчеркнул Попов, уже сейчас отечественное ПО поддерживает все

функции по администрированию АС «Выборы», в т.ч. ведение реестра избирателей и автоматизацию избирательных процессов. Однако полный переход на новую ОС — дело не одного дня, а ГАС «Выборы» должна работать безостановочно.

По мнению и.о. генерального конструктора ГАС «Выборы» Владимира Кабанова, разработка национальной ОС — реальная задача; на её решение уйдёт года два. При этом Минкомсвязи и ФГУП «Восход» (где спроектирована ГАС «Выборы») в порядке инициативы эту работу уже ведут.

Ещё в 2010 году Владимир Путин, как председатель правительства РФ, определил, что федеральные органы исполнительной власти и федеральные бюджетные учреждения должны перейти на СПО не позднее 2015 года, а утверждённая в том же 2010 году программа «Информационное общество» предусматривала создание национальной программной платформы к 2013 году. Но до сих пор ничего подобного не появилось.

РЖД + РФЯЦ = СИНЕРГИЯ

Объединили усилия

Реакторы и поезда перейдут на Linux?

РЖД совместно с Российским федеральным ядерным центром (РФЯЦ-ВНИИЭФ, г. Саров) займётся разработкой отечественной ОС «Синергия» на базе Linux и приложений на базе открытого ПО, сообщил замначальника департамента информатизации и корпоративных процессов управления РЖД Вадим Москаленко. Это решение приняли 13 августа 2014 г. на совещании руководства РЖД, «Росатома» и РФЯЦа, пояснил замдиректора РФЯЦ по ИТ Олег Кривошеев. Пилотный вариант ОС и системы управления базами данных (СУБД) уже написан и тестируется РФЯЦ, с целью отказаться от ОС Windows. РЖД двигались в том же направлении, и усилия решили объединить. Так как открытое ПО создается сообществом, речь идет о фиксации для нужд РЖД «определенной ветки развития соответствующего продукта», пояснил представитель РЖД. РЖД только доработает

функции для бизнес-приложений, отсутствующие в СПО. Сумму затрат на проект и число занятых в нем программистов РЖД и РФЯЦ не обнародовали.

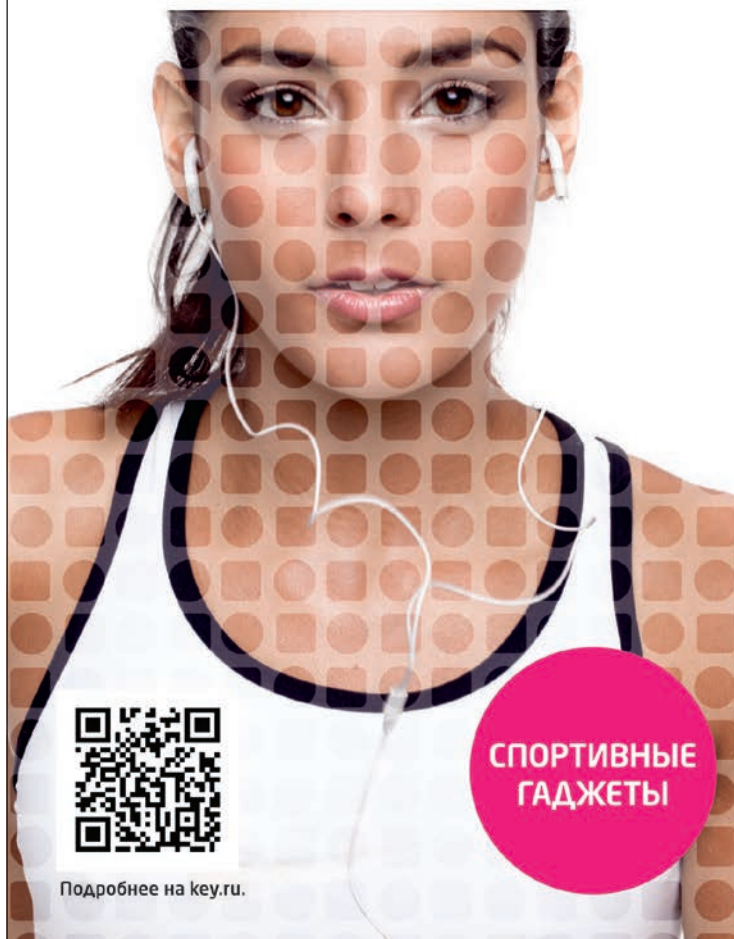
«Синергия» пока на начальной стадии, и ее перспективы не ясны, считает основатель компании «Новые облачные технологии» Дмитрий Комиссаров: для поддержки своего дистрибутива Linux нужна команда в 500 человек, по несколько на каждую подсистему, но в РФ сейчас такой нет. И РЖД располагает разработчиками прикладных программ, но не системного кода.

Планируется, что созданная платформа будет «отвечать современным требованиям к ИБ», сообщил представитель РЖД. Но наличие исходного кода — не гарантия безопасности, считает Илья Медведовский, гендиректор Digital Security, занимающейся анализом защищенности информационных систем: оценка надежности таких решений крайне сложна. **LXF**

Новости короткой строкой

- » К Linux Foundation присоединились компании Adapteva, GitHub, SanDisk, Seagate и Western Digital. Источник: www.linuxfoundation.org
- » Леннарт Поттеринг [Lennart Poettering] представил *systemd 216*. Источник: lists.freedesktop.org
- » Релиз *BricsCAD 14.2.14*, лидера САПР-платформ — альтернатив *DWG*, ввел для Linux модуль Communicator. Источник: www.bricsys.com/rss/
- » АНБ США и Центр правительственной связи Великобритании ищут уязвимости в исходных кодах *Tor*. Но доброжелатели извещают разработчиков о найденных ошибках, заявил лидер *Tor*. Источник: www.bbc.com/news/
- » 13–15 октября в Дюссельдорфе (Германия) пройдут мероприятия LinuxCon + CloudOpen + ELC Europe. Источник: events.linuxfoundation.org
- » Под GPLv3 вышла платформа совместной разработки Kallithea, от сообщества и Software Freedom Conservancy. Источник: lists.sfconservancy.org
- » Российская компания Eltechs выпустила Eltechs ExaGear Desktop — VM для запуска приложений для Intel x86 на мини-компьютерах на базе ARM. Источник: eltechs.com
- » *Viber*, приложение VoIP, и *MEGAsync*, клиент синхронизации данных с облачным хранилищем, получили для Linux ту же функциональность, что и для Windows и Mac OS X. Источник: www.viber.com, mega.co.nz
- » В Ubuntu нашли проблему в инсталляторе. При нескольких ОС на ПК переустановка могла уничтожить данные. Источник: www.omgubuntu.co.uk
- » Open webOS, открытая встраиваемая ОС, переименована в LuneOS и поддерживает HP TouchPad, Nexus 4, Galaxy Nexus и Nexus 7. Источник: www.engadget.com
- » Collabora разработала *Epiphany for Raspberry Pi*, оптимизированный для ARMv6, с поддержкой HTML5. Источник: www.raspberrypi.org
- » FSF включил в список полностью свободных дистрибутивов *libreCMC*, для встраиваемых устройств. Источник: www.fsf.org

ТВОИ ПОМОЩНИКИ В ТРЕНИРОВКАХ



Подробнее на key.ru.

СПОРТИВНЫЕ
ГАДЖЕТЫ

Экшн - камера



Экшн-камера Sony HDR-AS30

Smart - часы



Smart-часы Cookoo Watch

Smart - здоровье



Браслет Jawbone UP



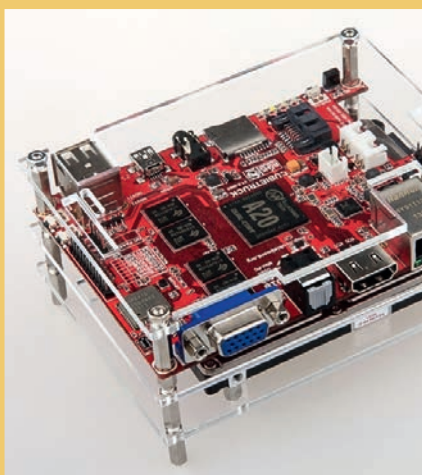
Шаромер Misfit Shine

Товар сертифицирован. Количество товара ограничено.



Одноплатный компьютер Cubieboard3

Cubietruck



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

- Процессор: AllWinnerTech SOC A20, ARM Cortex-A7 Dual-Core
- Видео: ARM® Mali 400 MP2
- Выходы: HDMI, VGA 1080p
- Оперативная память: 2 Гб, DDR3, 480 МГц
- Встроенная память: 8 Гб NAND
- Внешние накопители: интерфейс SATA 2.0 для 2,5" HDD; возможно подключение 3,5" HDD с использованием внешнего питания 12 В
- Сеть: 10M/100M/1G Ethernet, Wi-Fi и Bluetooth
- Питание: DC 5 В, 2,5 А с подключенным HDD; возможно подключение

литиевого элемента питания

- Периферия: 2 × USB host, 1 × USB OTG, 1 × Toslink (SPDIF), 1 × IR, 4 × LEDs, 3,5" аудиовыход, 3 кнопки
- Поддерживаемые ОС: Android, Linux-based, BSD
- 54 контакта расширения I2S, I2C, SPI, CVBS, LRADC × 2, UART, PS2, PWM × 2, TS/CSI, IRDA, LINEIN & FMIN & MICIN, TVIN × 4
- Размер платы: 11 × 8 см

КОМПЛЕКТАЦИЯ

- Плата Cubietruck
- Прозрачный акриловый корпус
- Кабель питания USB
- Кабель MiniUSB
- Кабель SATA с питанием
- Радиатор для процессора

4395
руб.

www.linuxcenter.ru/shop/ops_hard/cubieboard/cubietruck/

«ГНУ/Линуксцентр»: +7 812 309-06-86 (Санкт-Петербург), +7 499 271-49-54 (Москва)



Ключевое событие для HR-специалистов

29-30 октября

hrperezagruzka.ru

забронировать участие
можно также по телефону

+7 812 607 77 29



**HR-ПЕРЕЗАГРУЗКА
от HeadHunter**

hh **ru**
HeadHunter



Новинки программного и аппаратного обеспечения в описании наших экспертов



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

И снова о Linux'е и поэзии

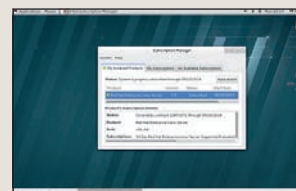
Тема взаимодействия Linux'a и творческой интеллигенции, в первую очередь литературной, уже поднималась в этой колонке (см. **LXF99** и **LXF100**, 2008 год). Ныне же появился повод вернуться к ней снова.

Повод этот — выход в электронном виде сборника стихов «В начале» Веры Хорват — сербской поэтессы (она же эссеист, литературный переводчик, художественный критик etc., обладательница всяческих поэтических регалий, лауреат, участник, и прочая, и прочая, и прочая). Это первая её книга, вышедшая в русском переводе, выполненном Олегом Комковым. Пересказывать стихи — занятие неблагодарное, и я от него воздержусь. Отмечу только, что ряд их посвящён не очень известным моментам древней истории, например, культуре Лепенски-Вир, существовавшей на территории Сербии 7–9 тысяч лет назад, или Фестскому диску, тайна письмен которого не разгадана до сих пор.

Требует объяснения, какое отношение чей-то поэтический сборник имеет к Linux'у. А самое непосредственное: его электронная версия была подготовлена с помощью исключительно свободного ПО, работавшего под управлением этой ОС: *LibreOffice* и дополнения к нему *OOoFBTools*. Без этой триады сборник был бы недоступен для народа: бумажная версия его стала библиографической редкостью ещё до выхода из типографии. Версию же электронную (в форматах PDF, FB2, HTML) можно найти в Библиотеке Блогосайта на <http://alv.me> и на всем известной Флибусте. alv@posix.ru

Сегодня мы рассматриваем:

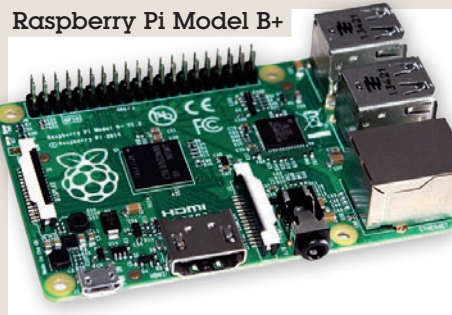
- RHEL 7** 14
Свежий релиз Красной Шапочки для предприятий, знаковый и для корпоративных, и для домашних пользователей, с отличными функциями вроде автоматического обновления без переустановки.
- Opera Developer 24** 18
Это версия для разработчиков и любителей свежатинки вносит в Linux новейшие функции браузера *Opera*, включая Stash, Speed Dial и Opera Turbo.
- КаОS 2014.06** 16
Независимый дистрибутив, сосредоточенный на элегантности, функциональности и удобстве пользователя.
- HP 255 G1** 19
Малобюджетный ноутбук с Ubuntu на борту. Станет ли он победителем конкурса по безыскусности?
- MaKuluLinux 6.0** 17
Этот дистрибутив не похож на другие и намеревается предлагать «целый мир возможностей»... а заодно и массу удобных пост-установочных скриптов.
- Raspberry Pi Model B+** 20
Прибавилось портов и контактов GPIO, и вся плата капитально переработана. Именно этого вы и ждали?



➤ При виде RHEL 7 админы серверов запрыгают от счастья.



➤ Для *Opera* это возвращение в Linux; меньше показного блеска, больше функций.



➤ Кто стрескал все пирожки с малиной? Небось Лес Паундер, он до них большой охотник.

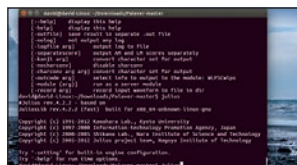
Сравнение: Распознавание речи

с. 24

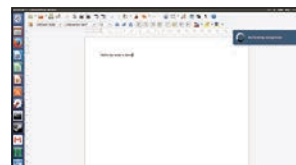
FreeSpeech



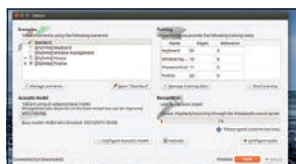
Julius



Palaver



Simon



Speech-App



Самые знаменитые приложения по распознаванию речи пробиваются в Linux с большим замедлением, но сейчас стали появляться отличные «родные» варианты. Мы протестировали пять лучших.

Red Hat Enterprise Linux 7

Маянк Шарма изучает последний релиз главного дистрибутива Red Hat, чтобы узнать, что задумали старшие пацаны и что это значит для нас.

Вкратце

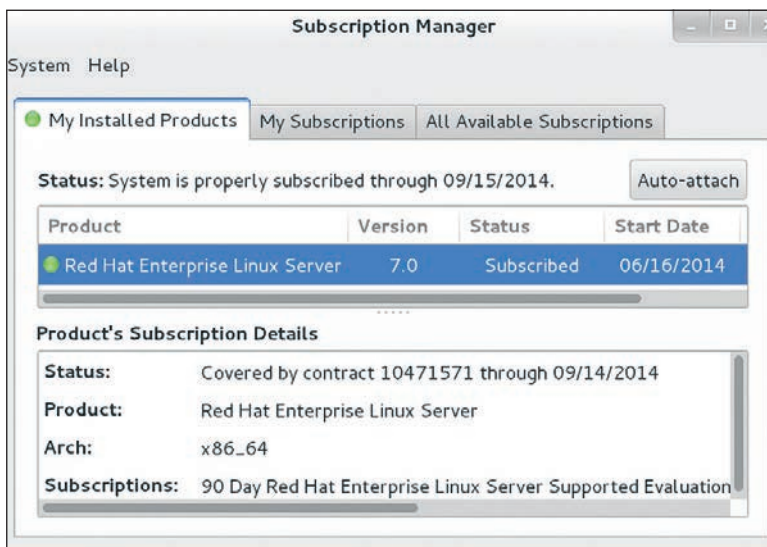
» Лучший корпоративный дистрибутив, который можно купить за деньги. См. также: SUSE Linux Enterprise Server.

Выход нового Red Hat Enterprise Linux — всегда большое событие для пользователей и корпоративного, и настольного Linux. Бесспорно, это самый популярный дистрибутив Linux для предприятий, сделавший из Red Hat компанию-миллиардера. Новый релиз обычно задаёт курс развития корпоративного Linux, опираясь на множество отпрысков, подобных CentOS, которые тоже по-своему популярны. Ну, а пользователям настольных систем новый релиз RHEL говорит о зрелости существующих технологий Open Source.

Последняя версия, RHEL 7, разрабатывалась довольно долго и даже провела более полугодия в стадии бета-тестирования, в ходе которой клиентам компании было предложено поучаствовать в тестировании и доработке ряда функций.

В плане характеристик, RHEL 7 знаменует собой отход от традиции. Для начала, файловой системой по умолчанию стала XFS, вместо ext4. Другие ключевые компоненты включают менеджер службы *Systemd* и загрузчик *Grub 2* с поддержкой UEFI и разнообразных платформ, в том числе архитектуры PowerPC.

В предыдущем основном релизе RHEL использовался демон инициализации



» RHEL 7 включает популярные графические оболочки, хоть они и ниже достоинства опытных сисадминов.

Upstart, ныне замененный на *Systemd*. Red Hat успел сгладить острые углы в новом менеджере загрузки, так как он уже несколько лет был в составе Fedora. Еще одно важное изменение касается ядра. Вы больше не сможете установить дистрибутив на старые 32-битные машины: отныне RHEL 7 поддерживает только 64-битные архитектуры, включая Intel x86-64, PowerPC и IBM s390.

Появление новой файловой системы по умолчанию также связано с переходом Red Hat на 64-битный вариант. XFS — 64-разрядная журналируемая файловая система, и одним из её достоинств является улучшенная производительность (особенно на больших файлах) и расширенная пропускная способность на многопроцессорных системах. Согласно Red Hat, XFS поддерживает файловые системы до 500 ТБ. Это позволяет дефрагментировать и расширять уже смонтированную, действующую файловую систему, а также обеспечить необходимое ей резервное копирование и восстановление объектов. Дистрибутив предлагает использовать *MariaDB*, но также поддерживает и *PostgreSQL 9.2*.

Вперед и вверх

Одним из главных улучшений в RHEL 7 является улучшенная совместимость и интеграция с *Active Directory* от Microsoft, которая позволяет Red Hat расширить своё присутствие в негомогенных по операционным системам многопользовательских средах. В RHEL 7 теперь можно установить перекрестное взаимодействие с *Active Directory*. Это дает пользователям с учетными данными *Active Directory* право доступа к ресурсам Linux без дополнительной аутентификации.

Всю тяжёлую работу выполняет новый компонент под названием *realmd*. Это сервис D-BUS, который автоматически обнаруживает информацию о домене и упрощает процесс настройки, необходимой для доступа к домену. Сервис работает с Microsoft *Active Directory* и Red Hat Enterprise Linux Identity Management.

Даже администраторы, которые выполняют автоматическое развертывание с помощью файлов *kickstart*, смогут автоматически интегрировать новые машины в домены *Active Directory*, создавая *kickstart*-файлы, не требующие прав администратора; тогда установленная система

Свойства навскидку

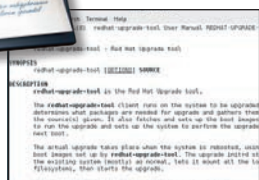


Оперативные обновления

Они были всеобщим пожеланием, и несмотря на ограничения, администраторы оценят их, в особенности при переводе своей инфраструктуры на новый релиз.

Контейнеры Docker

При запуске приложений в контейнерах, где ресурсами управляет ядро Linux, безопасность изолированных процессов — главная изюминка этого релиза.



сможет присоединиться к домену с одно-разовым паролем.

Ещё одна функция, которая наверняка порадует пользователей — возможность оперативных обновлений. Только в релизе RHEL 7 можно обновиться до последней версии, не снося существующую операционную систему. Однако этот метод имеет свои подводные камни. Его можно использовать только для обновления 64-разрядных установок RHEL 6 на платформах Intel (x86-64), хотя в будущих релизах планируется поддержка обновлений для PowerPC и архитектур s390x.

Кроме того, в процессе обновления обрабатывается ограниченное количество групп пакетов, включая Minimal, Base, Web Server, File Server, DHCP Server и Print Server. Обновления других групп и пакетов не поддерживаются, так что вам придется удалить их в RHEL 6 и переустановить на новой RHEL 7 вручную.

Процесс автоматического обновления обзавелся новым помощником — тот сканирует установленные системы и предоставляет отчет о том, что можно обновить автоматически, а что придется сделать вручную. Отчёт включает описание возможных проблем и ссылки на Базу

знаний в Портале пользователей Red Hat [Customer Portal]. Здесь также перечислены модифицированные пользовательскими файлами настройки и указано, которые из них будут изменены.

Red Hat использует причудливые эпитеты, чтобы подчеркнуть кроссплатформенность RHEL 7. Он может работать на чистых серверах, на виртуальной машине, а также

создания полноценной ОС, а вместо этого использует часть установленной. Преимущество этого подхода в том, что пользователи могут без труда перемещать свои Docker-приложения с одной платформы, скажем, чистого сервера, на другую, например, виртуальную машину.

Docker использует элементы ядра Linux, чтобы создать контейнер для программных приложений, находящихся на одном хосте, но выделяет каждому изолированный сегмент памяти, хранилища, процессора и сети. Это означает, что контейнеры создаются и запускаются быстрее, чем решения на базе гипервизора. Дэн Волш [Dan Walsh], инженер по безопасности Red Hat, говорит, что компания использовала Docker для запуска вплоть до 1000 сервисов Apache с одного узла.

В дополнение к установочным пакетам к Docker, Red Hat также планирует выпускать сертифицированные образы Docker, которые можно будет развернуть на RHEL 7. Пока что в репозиториях есть только образ базового сервера RHEL 7.

В составе RHEL 7 находятся инструменты из проекта OpenLMI, позволяющие администраторам управлять физическими и виртуальными серверами. Вы можете использовать их для настройки, управления и мониторинга аппаратных, ОС и системных сервисов. Сюда входит новый механизм оповещения об аппаратных событиях (hardware event reporting mechanism, HERM), представляющий информацию последовательно и в одном месте; кроме того, RHEL 7 позволяет в реальном времени, без перехода в автономный режим, осуществить миграцию виртуальных машин с RHEL 6 на новую версию.

Релиз RHEL 7 предлагает своим пользователям несколько причин для обновления и делает всё возможное, чтобы облегчить этот процесс. Тем не менее, компания выпустит также поддерживающий релиз RHEL 6, для тех, кто не собирается переключаться сразу. **LXF**

«Пользователям понравятся оперативные обновления.»

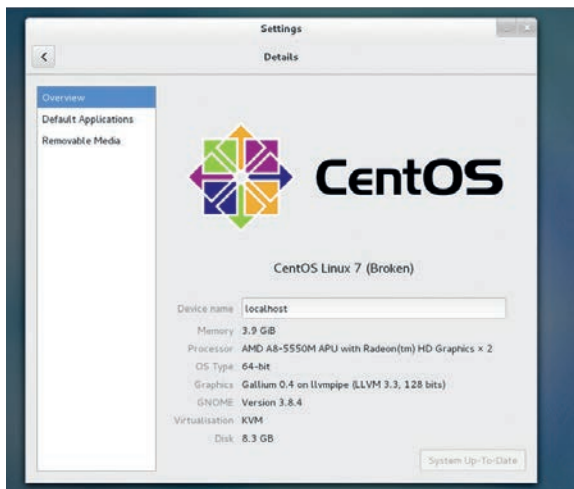
➤ Пользователи CentOS получают те же новые программы, что и у пользователей Fedora, сохранив стабильность RHEL.

в OpenStack-облаках типа Infrastructure-as-a-Service (IaaS) и Platform-as-a-Service (PaaS). Виртуализация в дистрибутиве по-прежнему реализуется на Kernel Virtual Machine (KVM), но представлена и новая контейнерная технология.

Сошлось

Контейнеры Linux — одно из главных событий этого релиза. Они обеспечивают реализацию песочницы для запуска отдельных приложений в безопасном контейнере изолированно от других, работающих на том же хосте. Контейнеры позволяют параллельно запускать нескольких независимых копий приложения, для совместного использования рабочего окружения и вычислительных ресурсов.

В этой связи, главное новаторство RHEL 7 заключается в интеграции Docker — виртуального приложения, приобретающего всё больше поклонников в корпоративной среде по всему миру. Это открытый инструмент, позволяющий упаковать приложение в виртуальный контейнер, который затем можно развернуть на самых разных Linux-серверах — нечто вроде гипервизора, но контейнер Docker легче, поскольку, в отличие от традиционной виртуальной машины, не требует



RHEL 7, CentOS и Fedora

Есть несколько проектов, которые берут GPL-лицензированный исходный код RHEL, отбрасывают бренд Red Hat и компилируют свой дистрибутив. Самый популярный корпоративный дистрибутив на базе RHEL — CentOS; он также повторяет родительские релизы и сейчас находится в своей седьмой итерации.

И хотя RHEL 7 питает CentOS 7, между проектами есть несколько различий. CentOS выпускает ядро CentOS Plus, где есть функции, отключённые в главном ядре, а также исправления известных проблем. В отличие от RHEL, CentOS 7 также будет доступен в виде live-носителей для установки. И хотя начальный релиз CentOS 7 предназначен для 64-разрядных машин

(также вслед за RHEL 7), проект CentOS планирует и релиз для i686.

Релиз RHEL 7 никак не отразился на другом спонсируемом Red Hat проекте, Fedora. Дело в том, что многие технологии RHEL 7 уже были в Fedora, выступающем для Red Hat в качестве инкубатора новых идей.

По словам экс-руководителя проекта Fedora Робин Бергерон [Robyn Bergeron], RHEL 7 во многом основан на Fedora 19, с элементами Fedora 20. Среди ключевых компонентов в RHEL 7, уже имеющихся в Fedora, менеджер сервисов Systemd, дебютировавший в Fedora 15, и установщик Anaconda со звездобразной структурой [hub-and-spoke], впервые введенный в Fedora 18.

LINUX **Вердикт**
FORMAT

Red Hat Enterprise Linux 7

Разработчик: Red Hat
Сайт: www.redhat.com
Лицензия: GPL и проприетарная

Функциональность	9/10
Производительность	9/10
Удобство использования	6/10
Оправданность цены	7/10

» Необходимый своим пользователям, RHEL 7 также склонит на свою сторону сомневающимися корпоративных клиентов.

Рейтинг 8/10

KaOS 2014.06

Дэвид Хейвард изучает интересный независимый дистрибутив, где удобство пользователя превыше содержания.

Вкратце

» Подтверждает, что создав дистрибутив с нуля, можно добиться действительно зрелищных результатов.

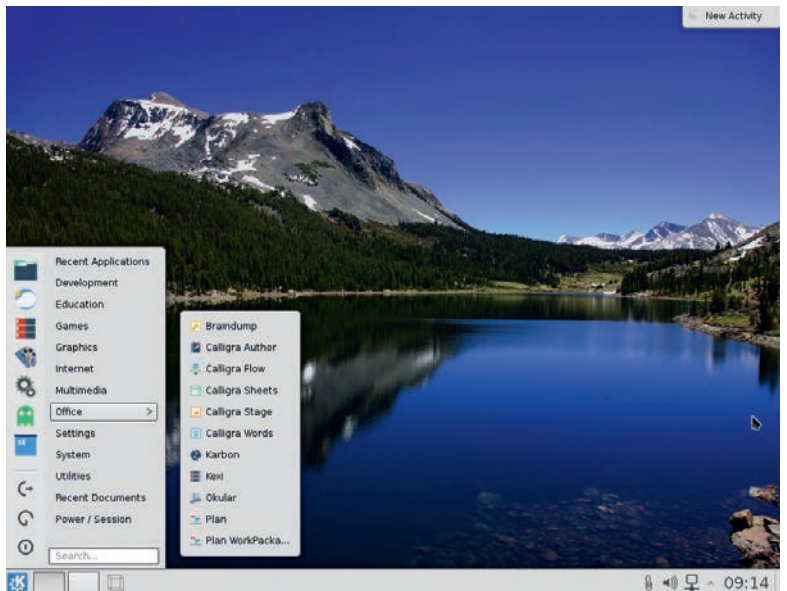
КaOS — это новый дистрибутив, ориентированный на элегантность и функциональность, а в плане удобства просто выдающийся.

Релиз 2014.06 — это поджарый дистрибутив на базе KDE и инструментария Qt, созданный с нуля, с обновлением по системе «плавающих релизов» и специально подобранным репозиторием. Упор делается на качество, а не количество, но выбор достаточно широк, и версии приближены к последним релизам: KDE 4.13.2, ядро Linux 3.14.6, *glibc 2.19*, *system 212* и *kmod 17*, а также *X.Org Server 1.15.1*, *Qt 4.8.6* и *QtWebkit 2.3.3*.

Что касается пакетов, к вашим услугам последняя версия web-браузера на базе Qt, *QupZilla*, и простой, но мощный *Octopi* в качестве GUI-интерфейса для *Pacman 4.1.2*, а также *KDE Telepathy* и *Kdenlive*; и *Calligra 2.8.3* в числе главных офисных приложений. Этого достаточно, чтобы сделать его интересной альтернативой наиболее популярных и широко используемых дистрибутивов.

Естественно, доступны и некоторые GTK-программы: *Firefox 29*, *Chromium*, плагин Flash и *GIMP 2.8.10* тоже присутствуют, но в урезанном варианте, в связи с принятой в репозиториях KaOS установкой на минимализм.

Те, кто уже пользовался KaOS в прошлом, будут рады узнать, что эта версия содержит много исправлений ошибок, изменений и улучшений по сравнению с предыдущими. Возможности медиа расширены за счёт интеграции *ffmpeg2* и добавления *Amarok* вместо *Clementine*, а также *Gstreamer 1.0*, пришедшего на смену *Gstreamer 0.10*.



» Чуть поднастроить, и эффект будет поразительный — но не для старых машин.

Недостатки руководства

Установщик выполнен с нуля и сравнительно проворен. Имеется в общем неплохой набор вводной документации, которая поможет новичку справиться с процессом установки и даже настройки и создания разделов на диске. Однако с момента, когда KaOS установлен, документация резко обрывается, что весьма осложняет задачу начинающему пользователю. Несколько простых советов по обновлению и установке программ стали бы хорошим дополнением к компьютеру.

KaOS, безусловно, стильный дистрибутив. Приятный, мягкий серый фон с абстрактными мазками цвета по всему экрану радует глаз. Системные шрифты крупные, легко читаются и щадят немолодые глаза. Это свежая, довольно шустрая и гибкая система, не перегруженная и лёгкая, пусть и не самая быстрая в загрузке и работе и более алчная до ресурсов, чем стандартный дистрибутив. Ладно; красота требует жертв. Но беда в том, что прямо из коробки KaOS навряд ли можно назвать идеальной настольной системой для повседневной работы. Конечно, это характерно для многих дистрибутивов, но здесь кое-какие ограничения в нём самом и в его репозиториях вызывают опасения, что KaOS может оказаться чересчур специфическим, а не просто стильной альтернативой именитых релизов.

Не то что разработчики должны вкидывать в ISO все старые программы подряд, но не мешало бы сделать набор пакетов по умолчанию более цельным или привычным, а также добавить пару пользовательских скриптов для новичков, подобно тому, как это сделали некоторые другие современные дистрибутивы.

В целом, KaOS 2014.06 — система Linux для истинных ценителей. Это умная сборка, из тех, которые нужно прочувствовать, а не наскоро проглотить и тут же отформатировать. Да, у неё имеются свои причуды, но в них можно даже найти известное удовольствие. **LXF**



Свойства навскидку



Свежий Pacman

Octopi, графический интерфейс к *Pacman*, мало весит, но многое может.



Свой установщик

Установщик весьма неплох и подчёркивает стиль и качество продукта.

LINUX Вердикт
FORMAT

KaOS 2014.06

Разработчик: Анке Берсма
 [Anke Boersma] и др
 Сайт: <http://kaosx.us>
 Лицензия: GPL

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

» Этот стильный дистрибутив можно смаковать как хорошее вино, но многое следует доработать.

Рейтинг 7/10

Макулунух 6.0

Интересный дистрибутив в непривычном стиле, который равно подойдёт и знатокам, и новичкам. Проверяет **Дэвид Хейвард**.

Вкратце

» Простой в использовании дистрибутив на базе Debian, с упором на стабильность, скорость и стиль, со множеством пакетов прямо из коробки.

Приложения

- » Chromium
- » Kingsoft Office
- » Kmail
- » PlayOnLinux
- » Deluge
- » GIMP
- » Скрипты настройки

Стараниями его главного разработчика, Жака Реме [Jasque Raymer], популярность MakuluLinux в последнее время стабильно растёт.

На зулусском языке Makulu означает «большой вождь»; дистрибутив существует в нескольких вариантах — с Mate, Xfce, Enlightenment и новейшей версией KDE. В каждом из них своё собственное уникальное окружение рабочего стола, но главной идеей Makulu является создание стабильного, быстрого и удобного дистрибутива, изначально доверху загруженного пакетами.

MakuluLinux предлагает пользователю не только красочный рабочий стол поверх стандартной сборки, но и кое-что для простоты и удобства работы. Например, при установке на жесткий диск можно выбрать между «классическим» режимом, с обычным набором вопросов и возможностью настроить систему на свой вкус с помощью GParted, и «новым», где количество задаваемых вопросов изрядно сокращено (до очевидных: имя пользователя, пароль и т.д.) и не надо возиться с GParted — это специально для непосвященных.

Далее следует ещё один этап, чтобы «Вождь» мог обрести полную силу. Это несколько максимально адаптированных пользовательских сценариев, которые предлагают несколько вариантов настройки системы. Такие опции, как Изменить пароль администратора, Определить расположение хранилища, Проверить наличие системных обновлений, Настроить dockmanager, Проверить наличие доступных драйверов и Установить дополнительное



» Рабочий стол MakuluLinux — красивая штука, и с чувством юмора.

рекомендуемое ПО помогут разобраться в дистрибутиве независимо от того, новичок вы в Linux или нет.

Полный комплект

По большей части, все эти сценарии выполняются идеально; единственно, что при выполнении пост-установочной настройки не запустилась наша проводная сетевая карта. Но, похоже, это общая проблема Debian, а не самого Makulu.

Сразу из коробки в Makulu установлено множество пакетов, так что вам будет чем поживиться. Было приятно увидеть PlayOnLinux, равно как и Chromium, Handbrake и GIMP. Но есть и странноватые решения — например, Kingsoft Office вместо LibreOffice; Kmail в качестве единственного почтового клиента, а также Popcorn Time (с сопутствующими ему подвохами в плане законности) и Deluge.

Кроме того, как мы уже отметили, есть пакет PlayOnLinux с залатанной версией Wine и полной поддержкой CMST, так что производительность в играх значительно улучшилась. Добавьте к этому клиент Steam, и получите симпатичный и очень достойный игровой ПК на Linux.

С точки зрения стабильности и скорости, у нас с Makulu не было никаких проблем. Система запускалась быстро, не тормозила и не падала внезапно, и прекрасно себя показала.

Лозунг MakuluLinux — «Целый мир возможностей», и здесь мы склонны

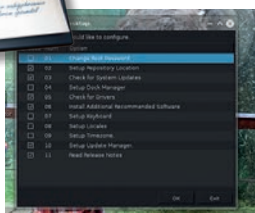
согласиться с разработчиками. Занятный дистрибутив, с массой всего интересного, идеальный вариант для новичка, а для разнообразия подойдёт и более опытным пользователям.

Другой вопрос, долго ли он пробудет вашим любимым дистрибутивом? В нём полно развлечений и программ, но может оказаться, что скоро вам это приестся и захочется чего-то более привычного, с менее специфическим оформлением.

Тем не менее, и с точки зрения самопровозглашенного дистрибутивомана, от MakuluLinux веет свежестью и новизной. Попробуйте этот любопытный дистрибутив — мы уверены, он вас не разочарует. **LXF**



Свойства навскидку



Пост-установочные скрипты

Безусловно удобны для настройки новичкам в Linux и работают эффективно.



Пакеты

В MakuluLinux полно предустановленных пакетов, хотя порой их выбор несколько удивляет.

LINUX FORMAT Вердикт

MakuluLinux KDE 6.0

Разработчик: Жак Реме и др.
Сайт: www.makululinux.com
Лицензия: GPL

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

» Очень занятный дистрибутив, только выбор пакетов из коробки иногда причудлив.

Рейтинг 8/10

Opera Developer 24

Марко Фиоретти испытывает крутой интернет-браузер, который впервые с июля 2013 года возвратился к ядру Linux.

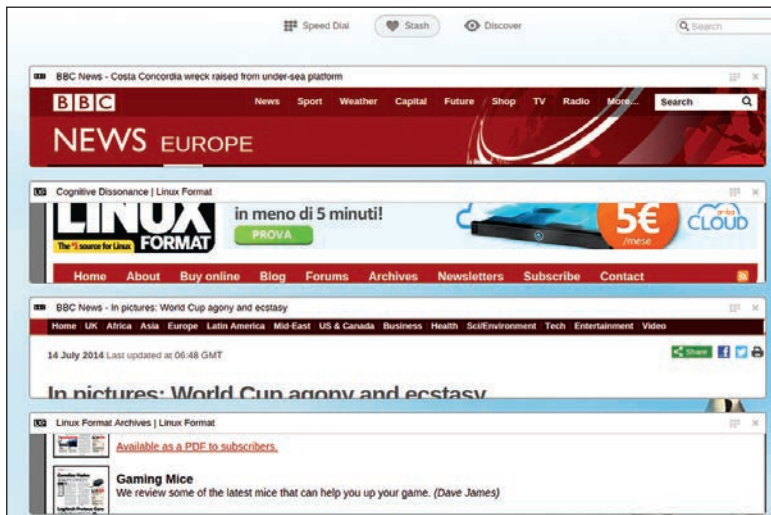
Вкратце

» Opera — полнофункциональный, легкий и быстрый браузер. Может помочь более эффективно работать в Интернете, даже при ограничениях, немного досадных пользователям Linux.

Opera Developer — версия браузера Opera для разработчиков и любителей экспериментов. В июне 2014 года Opera вернулся к Linux с браузером Developer 24 на базе Chromium. Естественно, мы тут же его загрузили. На время написания статьи Linux-версия была доступна только как Deb-пакет, «испытанный на 64-битной Ubuntu с Unity или Gnome Shell», и мы опробовали его на Ubuntu 14.04. Обнаружился чистый, интуитивный интерфейс, отзывчивый даже при ОЗУ на 1-2 ГБ. Наведение курсора на заголовок вкладки высветило предпросмотр ее содержания, достаточно обширный, чтобы проверить изменения, не выходя со страницы. Также есть модуль импорта закладок и контактов из Explorer, Outlook и Firefox; «качания» мыши позволяют скакать по страницам вперед-назад. Официальные расширения включают менеджеров конфиденциальности, объединенные интерфейсы Gmail и Evernote, генераторы QR-кода и виджеты для Speed Dial (см. ниже). Вполне приличная документация доступна онлайн.

Здесь мы подробно остановимся на Discover, Stash, Speed Dial и Opera Turbo. Stash предназначен для быстрого доступа к произвольной странице, но не всегда. Щелкните правой кнопкой на любом месте страницы и выберите Add to Stash. (Удалите это, когда пожелаете.) Значок-лупа вверху справа открывает истинный смысл Stash — это функция поиска, сканирующая метаданные всех таких страниц, включая описания и тэги, встроенные в их HTML-источники. Приборная панель Speed Dial позволяет упорядочить динамические предпросмотры чаще всего вызываемых важных сайтов: можно открыть любую из их миниатюр, лишь напечатав ее номер и нажав Enter. Она еще умеет поддерживать динамический контент, типа уведомлений email, через расширения.

» В Opera Speed Dial по умолчанию есть ряд поисковых алгоритмов. На практике возможны и другие.



» Stash предлагает страницы, важные для вас именно сейчас, в одном доступном для поиска перечне.

Speed Dial щеголяет стационарной, броской поисковой строкой Google. Допускаются другие поисковики по умолчанию, но только в верхней поисковой/адресной строке и только с одобрения Opera; таковыми являются Amazon, Bing, Yahoo! и Wikipedia. Очевидно, это предотвращает перенаправление вредоносным ПО ваших поисковых запросов к серверам жуликов. Можно добавить другие движки поиска, но помните, что сначала нужно впечатать их сокращенное наименование в поисковую/адресную строку.

Скоростной серфист

При работе в Turbo-режиме браузер запрашивает у серверов Opera отправку сжатых копий страниц, которые надо просмотреть. Opera утверждает, что можно загружать страницы «до пяти раз быстрее, чем другие браузеры». Однако безопасные URL (https) будут доступны напрямую, чтобы не пропускать конфиденциальные данные через те самые серверы.

Таблица Discover внешне очень похожа на блог Tumblr рекомендованного web-содержания. Кроме вида, реальная разница между Discover и RRS — в «рекомендуемой» части. Можно выбрать интересные вас страны и избежать просмотра всех категорий, но в любом случае нельзя изменить predetermined перечень источников новостей.

Для пользователей Linux Opera Developer 24 имеет некоторые ограничения.

Прежде всего, вне Ubuntu он может не работать. Далее, ряд его частей по природе неизменяемы. Установленный по умолчанию Speed Dial, при обязательном окне Google с толпой логотипов, похивает признаками скатывания Интернета к кабельному ТВ — без нейтральности сети. Закрепленный перечень источников Discover и того хуже, но «принуждение» к изучению определенных мнений не всегда бывает плохой идеей.

Для многих пользователей Opera Developer 24 по крайней мере не хуже Firefox или Chrome, а часто и быстрее. А Stash великолепен. Поисковик ограничивается страницами, важными для вас в данный конкретный момент? Да и отлично же! **LXF**

LINUX FORMAT Вердикт

Opera Developer 24

Разработчик: Opera Software
Сайт: www.opera.com
Лицензия: Бесплатное ПО

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

» Крутой браузер. Если вы не против его выбора поддерживаемых дистрибутивов, новостей и поисковых движков, попробуйте!

Рейтинг **9/10**

HP 255 G1

Оснащенный Ubuntu, бюджетный вариант ноутбука от больших парней из HP по карману даже **Нейлу Мору** — но при такой производительности, оно ему надо?

Вкратце

» Малобюджетный, мало-мощный ноутбук поставляется с предустановленной Ubuntu 12.04 LTS. Идеален для бизнеса и обучения, но не исключено, что вам захочется HP Chromebook 11 или Dell Chromebook.

Спецификация

- » 2-ядерный AMD E1-1500 1,48 ГГц
- » DDR3 4 ГБ
- » HDD 750 ГБ
- » Дисплей: 15,6 дюймов 1366×768
- » Графика: AMD Radeon HD 7310
- » ОС: Ubuntu 12.04 LTS
- » Сеть: Gigabit, 802.11n
- » Оптический привод: DVD RW +/-
- » Дополнения: web-камера, Bluetooth 4 + HS
- » Порты: VGA, HDMI, 3×USB, слот SD

Настольный Linux обсуждают редко, еще реже перепедают ноутбуки с ним, но вот вам модель HP с предустановленной Ubuntu; на ней даже красуется наклейка Ubuntu. Судя по цене (менее £220), это явно бюджетный вариант. Что не обязательно означает плохое качество: мы думаем, он больше ориентирован на компании и школы, чем на домашних пользователей Linux.

Первое впечатление хорошее. Хотя дизайн призов не урвет, пластиковый корпус довольно прочен. Впрочем, как нам кажется, обшарпается он скоро, а шарниры стоило сделать чуть понадежнее. Приятный момент — традиционная сумка для ноутбука в комплекте.

Первая загрузка — обычный процесс установки Ubuntu. Странно, что он не иницирует соединение с беспроводной сетью, т.е. обновления не загружаются, пока вы не подключитесь сами. Загрузившись, вы обнаружите весьма обычную Ubuntu 12.04 LTS. После тестирования мы загружаем предусмотренное обновление LTS 14.04 (неофициальное до выхода знаковой версии 14.10). Все вроде гладко, но... по дороге грохнулись тачпад, драйверы Ethernet и Bluetooth. Клавиатура крепкая и отзывчивая, без отдачи. В HP не пытались особо умничать, и уцелела даже клавиша Windows — как напоминание, что вам повезло с Linux. Тачпад отзывчивый, и при нажатии кнопки происходит четкий щелчок. Ряд крохотных индикаторов показывает наличие питания, работу HDD и сети и статус беспроводной активности.

Наша основная критика касается дисплея. Мы вовсе не против разрешения 1366×768 — по факту, оно все еще имеет даже в моделях Apple низшего уровня;



» Смотрится простенько, но на худой конец вышел с наклейкой Ubuntu.

да и воспроизведение цветов однородное; но дешевая LED-панель имеет малый угол обзора и явный вертикальный градиент при отображении темных фонов.

Ноутбук использует спаренные беспроводной Ralink 802.11n и адаптер Bluetooth 4.0. Мы вначале нашли это соединение очень медленным; одиночная беспроводная антенна тоже здорово тормозила. Bluetooth работал хорошо, без вопросов соединяясь с нашими наушниками.

Медленно, но верно

За производительность отвечает интересный процессор AMD DC E1-1500. При двух ядрах Bobcat на 1480 МГц он не особенно мощный: всего лишь обогнал более старый Intel Atom D525 и обеспечивает половину скорости Intel Celeron 2955U, применяемого в настоящее время многими нетбуками Chromebook. Другой ограничивающий фактор — одноканальная шина памяти; но она хотя бы 64-битная, со всеми преимуществами 4 ГБ памяти DDR3.

Разочаровывает (как минимум, технически) втиснутая в процессор Radeon HD 7310 GPU. Шейдеры Radeon способнее микросхем той же ценовой категории, но это не говорит о многом в 2014-м, ибо при своей стоимости предлагает более низкую производительность. Что это значит? Нельзя сыграть в *Amnesia: The Dark Descent*, а о *Metro: Last Light* и не мечтайте.

В целом пользование ноутбуком было чертовски четким, несмотря на наши сожаления о медленном процессоре. Все прошло спокойно, даже загрузка, и ощущалась приятная отзывчивость. Накопитель на 750 ГБ разбит на раздел восстановления, основной раздел и раздел подкачки, но здесь и пространства с избытком. Также отметим, что отобранный BIOS раздел восстановления работает безупречно. Батарея средняя, на 4300 мА·ч, и на нашем ненаучном общем исследовании выдюжила 3,5 часа. Хвастаться особо нечем, но в режим приостановки работы и засыпания входит без вопросов. **LXF**

Сравнение навскидку

Тест	Zbox Nano	NUC DC33217	NUC D54250	HP 225 G1
Blowfish*	4,726	5,846	4,725	17,2
Cryptohash**	327,085	246,524	326,134	38,168
Fibonacci*	1,669	2,552	1,587	6,025
N-Queens*	7,346	6,438	5,082	25,684
FFT*	1,011	1,581	0,992	8,743
Raytracing*	6,202	7,826	6,198	16,658

* Меньше — лучше. ** Больше — лучше.

LINUX Вердикт
FORMAT

HP 255 G1

Разработчик: HP
 Сайт: <http://bit.ly/LXFhp255>
 Цена: £ 220

Функциональность	7/10
Производительность	4/10
Удобство использования	9/10
Оправданность цены	9/10

» Надежный ноутбук Ubuntu, но чтобы перешибить Chromebook, ему не хватает амбиций и скорости.

Рейтинг 7/10

Raspberry Pi B+

Как фаната всего, что связано с Pi, Леса Паундера одним из первых допустили к новому Raspberry Pi Model B+. Чем же этот Pi отличается от предшественников?

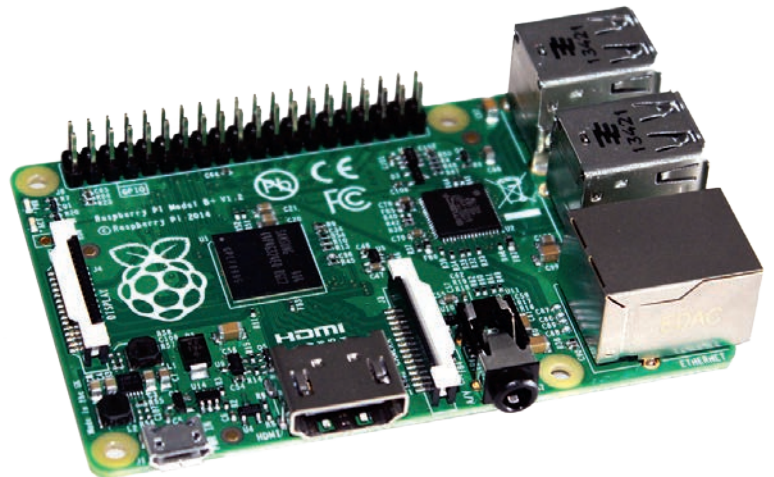
Вкратце

» Одноплатный компьютер, созданный для вдохновения и обучения, с недорогими, но доступными компонентами, чтобы им мог воспользоваться кто угодно.

По горячим следам Compute Module [См. Обзоры LXF187, стр. 18], Raspberry Pi Foundation неожиданно решили пополнить свой каталог ещё одним одноплатным компьютером. Raspberry Pi Model B+ — последний в серии Model B. (Аналогичный вариант A ожидается позже в этом году.)

Как и предыдущие модели, B+ использует систему-на-чипе (SoC) от Broadcom, дополненную BCM2835, процессором ARM11 с частотой 700 МГц и 512 МБ ОЗУ, однако, в соответствии с пожеланиями сообщества, улучшился макет. Теперь порты не разбросаны по четырём сторонам, а сконцентрированы на двух, с добавлением ещё 2-х дополнительных USB 2.0, итого их четыре. Из-за них-то и понадобилась модернизация. Рядом с USB расположен порт Ethernet, как и на предыдущей модели B, и далее по кругу — разъем для наушников, теперь с аналоговым аудио- и видеовыходом в четырехполюсном варианте. Это позволяет отказаться от отдельного композитного видеовыхода и экономит пространство. Аналоговый аудиовыход также усовершенствован. Пропуская стандартный порт HDMI, последний — Micro USB, подключен к более эффективной цепи питания, что уполовинивает энергопотребление всего до 1 Вт — мы думаем, что это существенно продлит жизнь любых проектов с питанием от батарей. Другое усовершенствование расположено снизу: вместо слота SD-карты появился тактильный слот MicroSD.

Возможно, вы заметили, что GPIO (General Purpose Input Output) также на вид



> Модель B+ немного меньше и имеет улучшенный макет с добавочными USB-портами и контактами.

Спецификация

- » 40-контактный GPIO
- » 4×USB 2.0
- » microSD
- » Broadcom BCM2835 SoC
- » CPU ARM 1176JZF-S 700 МГц
- » GPU VideoCore IV 250 МГц
- » ОЗУ 512 МБ
- » 100-МБ Ethernet
- » HDMI, RCA, гнездо 3,5 мм

немного увеличился: Foundation добавили туда ещё 14 контактов, увеличив их общее число до 40. Из этих 40 контактов, первые 26 полностью совместимы с исходным GPIO Raspberry Pi, а значит, большинство плат расширения будет работать с B+. Например, мы успешно протестировали его с PiBrella и PiGlow от Pimoroni. Правда, у нас возникли проблемы с обеими Wolfson Audio, поскольку в B+ отсутствует выход P5, необходимый для их подключения; но популярная PiFace должна быть совместима с Model B.

Больше контактов

Джеймс Адамс [James Adams], глава отдела по разработке оборудования в Raspberry Pi Foundation, подтвердил, что платы расширения, разработанные специально для B+, могут не стыковаться с предыдущей Model B, но пока нет тех, что употребляли бы все 40 контактов нового GPIO.

Дополнительные контакты дают SoC новые возможности — теперь их хватит на крупные проекты; кроме того, с помощью двух новых контактов GPIO — 27 и 28 — в будущем платы смогут использовать чип EEPROM для автоматической настройки системы при загрузке.

Все программные проекты, например, ScratchGPIO или Python, созданные на предыдущем Raspberry Pi, будут также полностью совместимы с B+, так что портировать проекты будет не сложнее, чем скопировать файл.

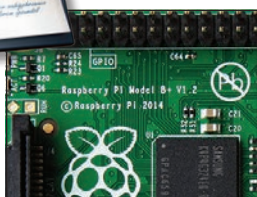
Очевидно, что B+ может больше предложить пользователям Raspberry Pi, но это

не значит, что выпуск модели B скоро прекратят. Адамс подтвердил нам, что до тех пор, пока есть спрос, Фонд будет продолжать их производство. Также будут выходить изменения и добавления ПО для обеих моделей.

При том, что у обеих точно есть будущее, в модели B+ исходный дизайн изменён в лучшую сторону. Это не даёт никаких существенно новых возможностей, но значительно повышает её потенциал, благодаря расширенному GPIO и дополнительным USB-портам, да ещё и за меньшие деньги и с более низким энергопотреблением. Хотелось бы, конечно, увидеть встроенный Wi-Fi, Bluetooth, USB 3.0, более быстрый CPU/GPU и больше памяти, но это нарушит целостность платформы, а она важнее всего. LXF

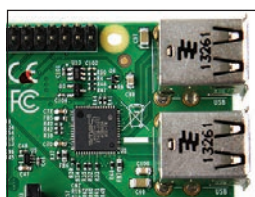


Свойства навскидку



Расширенный GPIO

Число контактов в GPIO увеличилось до 40. Два из них предназначены для автоматической настройки плат на базе EEPROM.



Больше USB-портов

Число портов USB 2.0 теперь увеличилось до четырех, и в B+ введена поддержка оперативной замены устройств.

LINUX FORMAT Вердикт

Raspberry Pi Model B+

Разработчик: Raspberry Pi Foundation
Сайт: www.raspberrypi.org
Цена: £25

Функциональность	8/10
Производительность	6/10
Удобство использования	8/10
Оправданность цены	10/10

» Долгожданная B+ с добавлением новых портов и контактов несёт сообществу желаемые перемены.

Рейтинг **7/10**

Добро пожаловать в робототехнику!

ScratchDuino

Электронный комплекс на основе свободного аппаратного обеспечения для школ и вузов

ScratchDuino.Лаборатория

Плата расширения с датчиками ввода-вывода — как установленными на плате, так и подключаемыми. Предназначена для программирования взаимодействия компьютера с внешними устройствами.

Полная интеграция («из коробки») со средой программирования Scratch, предназначенной для детей.

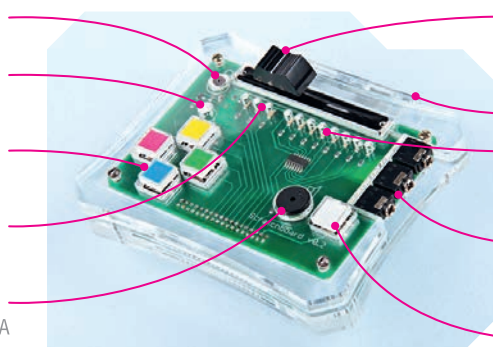
ДАТЧИК ЗВУКА*

ДАТЧИК ОСВЕЩЕННОСТИ*

КРЕСТОВИНА КНОПОК

ЦВЕТНЫЕ СВЕТОДИОДЫ

ДИНАМИК ДЛЯ ВЫВОДА ЗВУКА



ПЕРЕМЕННЫЙ РЕЗИСТОР (ПОЛЗУНОК)*

КАРТРИДЖ ARDUINO

СВЕТОДИОДЫ, УПРАВЛЯЕМЫЕ ЧЕРЕЗ РАЗРЯДНУЮ МАТРИЦУ

РАЗЪЕМЫ ДЛЯ ПОДКЛЮЧЕНИЯ ВНЕШНИХ ДАТЧИКОВ

КНОПКА*

* РАБОТАЮТ В РЕЖИМЕ ЭМУЛЯЦИИ RISCBOARD, НЕ ТРЕБУЯ ДОПОЛНИТЕЛЬНОГО ПРОГРАММИРОВАНИЯ.

ЦЕНА **5000 руб.**

ГАРАНТИЯ 3 ГОДА, ВКЛЮЧАЯ ТЕХНИЧЕСКУЮ И МЕТОДИЧЕСКУЮ ПОДДЕРЖКУ

ScratchDuino.Робоплатформа

Внешний робот-исполнитель, управляемый из среды программирования Scratch. Не требует навыков программирования на языках высокого уровня и может применяться для обучения, начиная с младших классов.

ScratchDuino.Робоплатформа

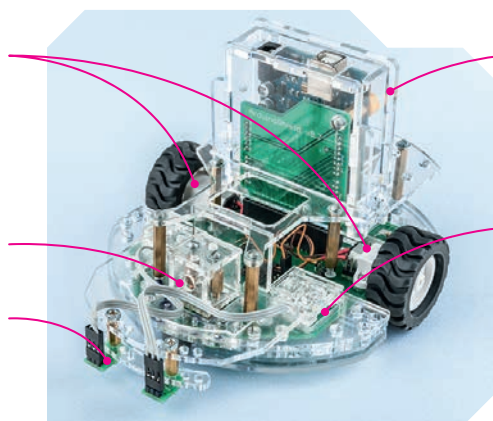
обеспечивает возможность:

- подключения внешних датчиков без механических соединений и пайки;
- расширения через последовательную шину RoboBus;
- установки деталей Lego Technics.

МОТОР-РЕДУКТОРЫ

ФОТОДАТЧИК

ДАТЧИК ЛИНИИ



КАРТРИДЖ ARDUINO

МАГНИТНЫЙ ЦИФРО-АНАЛОГОВЫЙ РАЗЪЕМ ДЛЯ ПОДКЛЮЧЕНИЯ ВНЕШНИХ ДАТЧИКОВ БЕЗ МЕХАНИЧЕСКИХ СОЕДИНЕНИЙ И ПАЙКИ*

* ВСЕГО НА РОБОПЛАТФОРМЕ 5 РАЗЪЕМОВ. ВОЗМОЖНО ПОДКЛЮЧЕНИЕ ДАТЧИКОВ КАСАНИЙ, ИК-ДАТЧИКОВ РАССТОЯНИЙ/ПРЕПЯТСТВИЙ И ДР.

ЦЕНА **10 000 руб.**

ГАРАНТИЯ 3 ГОДА, ВКЛЮЧАЯ ТЕХНИЧЕСКУЮ И МЕТОДИЧЕСКУЮ ПОДДЕРЖКУ

Назначение ScratchDuino

- Изучение взаимодействия компьютера с внешней средой
- Изучение изменения параметров внешней среды
- Изучение процессов передачи информации и принципов ее построения
- Изучение внешних устройств управления
- Моделирование устройств

Среда разработки

Lazarus (язык Pascal) ■ Scratch ■ Arduino IDE

ScratchDuino адаптирован для учебных заведений. Поставляется с комплектами учебно-методических материалов.

Продукт разработан при финансовой поддержке Фонда содействия развитию малых форм предприятий в научно-технической сфере

WWW.SCRATCHDUINO.RU

(812) 309-0686 (Санкт-Петербург) ■ (499) 271-4954 (Москва)



НОВОСТИ Плюсы и минусы фрагментации » Берегите данные » Замах на 64 бита » Путешественникам » Cyanogen бросает вызов Google » Планшет от Lenovo

ПУСТЬ РАСЦВЕТАЕТ СТО ЦВЕТОВ

Фрагментация: за и против

Фрагментация — и главная проблема, и основное достоинство Android.

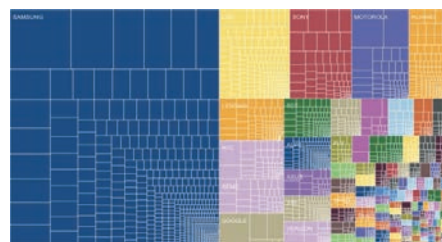
Фрагментация — и достоинство, и недостаток экосистемы Android: будучи головной болью для разработчиков, она зато обеспечивает глобальный охват рынка мобильных устройств. Такой вывод делает OpenSignal, британский стартап в области мобильного краудсорсинга, известный приложением для улучшения качества приёма мобильного сигнала.

Устройства на Android имеются на любой вкус, любых форм и размеров, и различаются по производительности и размерам экрана. Плюс, масса различных версий Android активны одновременно, добавляя еще один уровень фрагментации. В этих условиях разработка приложений чрезвычайно

сложна. Но дешевизна телефонов на Android (хотя бы и не с самой новой версией) означает более широкую аудиторию против устройств на iOS. Таким образом, именно благодаря фрагментации Android доминирует среди мобильных ОС.

За несколько месяцев 2014 года приложение от OpenSignal загружали 18 769 различных Android-устройств — рост на 60% против 11 868 в 2013-м, а против 2012-го более чем вчетверо — тогда было «всего лишь» 3997. Но в 2013 году 10 самых популярных Android-устройств занимали 21% рынка, а в 2014 году их доля упала до 15%.

Основной производитель Android-устройств — Samsung: ей принадлежит 12 из 13 наиболее



» Фрагментация Android по брендам, август 2014 г.

популярных, которые ныне занимают 43% рынка (чуть меньше прошлогоднего показателя 47,5%). Второе место — у компании Sony, но всего с 4,8%.

ОСТОРОЖНО: ВИРУС

Данные в опасности!

Google следует усилить безопасность платформы Android.

Минувшее лето было для Android не лучшим периодом. По статистике, 4 из 5 мобильных устройств работают сейчас под Android, и такая популярность подстрекает хакеров к выявлению уязвимостей, не присущих другим мобильным ОС. Вредоносных программ уже около 40 тыс. (для ОС от Windows и Apple таковых значительно меньше); среди них — SMS-

червь, способный распространяться. Производители не всегда вовремя обновляют версии ОС: дешёвые модели обычно «привязаны» к конкретной версии, и в эксплуатации остаются ненадежные старые версии. Анализ устройств на Android, купленных на eBay, показал: хотя перед продажей прежние владельцы удаляли личные данные, их удается восстановить. Из-за дефекта шифрования

обнаружились довольно откровенные селфи. Не лучше и сброс к заводским настройкам (та же проблема — у планшета Tesco Hudl, тоже на Android). Приложение Gmail допускает кражу данных. Тем же страдают мобильники от Microsoft и Apple, но до размещения в магазинах приложений их продукты жёстко проверяются, а Google Play — не единственный источник ПО для Android.

ТЕЛЕФОНЫ ОБГОНЯЮТ КОМПЬЮТЕРЫ?

Даешь 64 бита!

На подходе 64-битные мобильные процессоры для Android L.

Компания Nvidia приобрела лицензию на ARM v8, новейшую версию микропроцессорных команд от ARM, ещё в 2011 г. Наибольший интерес тут вызывал тот факт, что ARM v8 — это 64-битный набор. Создав 32-битные мобильные процессоры, Nvidia решила выйти на новый уровень: в платформе Tegra K1, выпущенной в начале 2014 года, планировалась

поддержка 64-разрядных процессоров для мобильных устройств. На конференции Hot Chips в Калифорнии представители Nvidia озвучили подробности о проекте, получившем кодовое имя Project Denver. Новый чип основан на 32-битной версии Tegra K1 и содержит 192 ядра архитектуры Kepler, при одинаковом с прежними Tegra расположением выводов. На одном кристалле размещены

2 ядра с частотой 2,5 ГГц и 7-канальной суперскалярной микроархитектурой (до 7 параллельных микроопераций за 1 такт), против 3-канальной у предыдущей версии. У нового процессора 4-канальный кэш данных L1 на 64 КБ и 16-канальный кэш L2 на 2 МБ. Nvidia рассчитывает обогнать существующие процессоры, реализовав поддержку Denver в ОС Android L уже в этом году.

К РАДОСТИ АВТОМОБИЛИСТОВ

Как проехать в библиотеку?

Обновлен Навител Навигатор 9.1 для Android.

Российская компания ЗАО «Центр навигационных технологий», разработчик навигационного ПО и навигационных карт, обновила программу Навител Навигатор для ОС Android.

Что нового, помимо ряда мелких исправлений, повышающих стабильность и надежность работы программы, найдут пользователи в версии 9.1.0.0?

Реализован учет дорожных ограничений по габаритам, массе и прочим параметрам транспортного средства, заданным пользователем при построении маршрута, для следующих категорий транспорта: Автобус, Спасательная служба, Служба доставки, Грузовой автомобиль. Определенные возможности проезда требуют наличия в картах информации о действующих на данной трассе

ограничениях. Исправлены ошибки, вызывавшие при построении маршрута образование «петель» (но эта функция будет доступна только с картами новой версии Q2 2014), и некорректное отображение пройденного маршрута на карте, а также ошибка, приводившая к утечке памяти при построении маршрутов между странами (при включенной услуге «Навител.Пробки»). Пользователям отныне доступно обновление карт России, Украины, Беларуси и Казахстана релиза Q2 2014. Карта России содержит уже 154676 городов и населенных пунктов, из них — 10487 с адресным поиском и улично-дорожной сетью. Адресная база достигла 8156640 уникальных адресов, а протяженность дорожного графа — 3870732 км. Обновлено



➤ На карте отмечены значимые объекты. Здесь: здание Варшавской оперы.

база Полезных Объектов Инфраструктуры: теперь в карте 1053187 POI. Улично-дорожная сеть приведена к состоянию на 1.08.2014 г. На карту России добавлены Крымский федеральный округ и детальные карты 64 городов.

ДРОЖИ, GOOGLE!

Знаки внимания к CyanogenMod

Проектом заинтересовались производители устройств на базе Android.

По мнению экспертов, развиваемая стартом Cyanogen прошивка способна стать полноценной альтернативой мобильной ОС от Google. Целый ряд известных компаний, занимающихся продажей устройств и приложений на Android и при этом не желающих находиться «под каблуком» у Google, обращают на неё всё более пристальное внимание. В CyanogenMod, альтернативной прошивке для устройств на Android, реализованы кодек для FLAC, клиент OpenVPN, меню перезагрузки и прочая функциональность, изначально отсутствовавшая в официальной версии мобильной ОС от Google. Android — ОС с открытым исходным кодом, но Google способна

влиять и на пути развития рынка аппаратных средств, и на используемое этой платформой ПО.

Вокруг CyanogenMod сложилось сообщество верных поклонников, и тысячи разработчиков со всего мира пишут для неё код; они полагают, что продукт Cyanogen уже вполне готов заместить Android. В минувшем году компания получила \$30 млн от венчурных фирм Andreessen Horowitz, Redpoint Ventures и Benchmark Capital — помимо средств Tencent, крупнейшей телекоммуникационной компании Китая. Издание *The Information* утверждает, что к развитию CyanogenMod с перспективой партнёрства или даже приобретения этого проекта проявили интерес Amazon.com, Microsoft,



➤ Сооснователи Cyanogen Стив Кондик [Steve Kondik] (слева) и Керт Макмастер [Kirt McMaster].

Samsung Electronics и Yahoo. По данным издания, представители Cyanogen встречались с и.о. гендиректора Microsoft Сатьей Наделлой [Satya Nadella].

НОВЫЙ ПЛАНШЕТ

Продукт Lenovo и Intel

Компания Lenovo выпустила свой первый Android-планшет с процессором от Intel.

До конца 2014 г. Lenovo откупит у Google компанию-производителя смартфонов Motorola Mobility (за \$2,91 млрд), после чего её смартфоны выйдут на рынок США. 3 сентября, в преддверии 543-й Международной выставки потребительской электроники IFA 2014, руководитель отдела потребительского маркетинга компании Lenovo Эндрю Бэрроу [Andrew Barrow]

представил новый планшет Tab S8 на Android — с 4-ядерным процессором Intel Atom с тактовой частотой 1,83 ГГц, с 8-дюймовым экраном 1920×1200 пикселей, двумя фронтальными динамиками, основной камерой на 8 Мпикс и фронтальной на 1,6 Мпикс. Весит он 10,5 унций (~280 г) и предлагается в корпусе белого, черного, желтого или синего цвета. Опционно поддерживается

стандарт LTE. По плану, старт продаж Tab S8 — во второй половине сентября, по цене \$199.

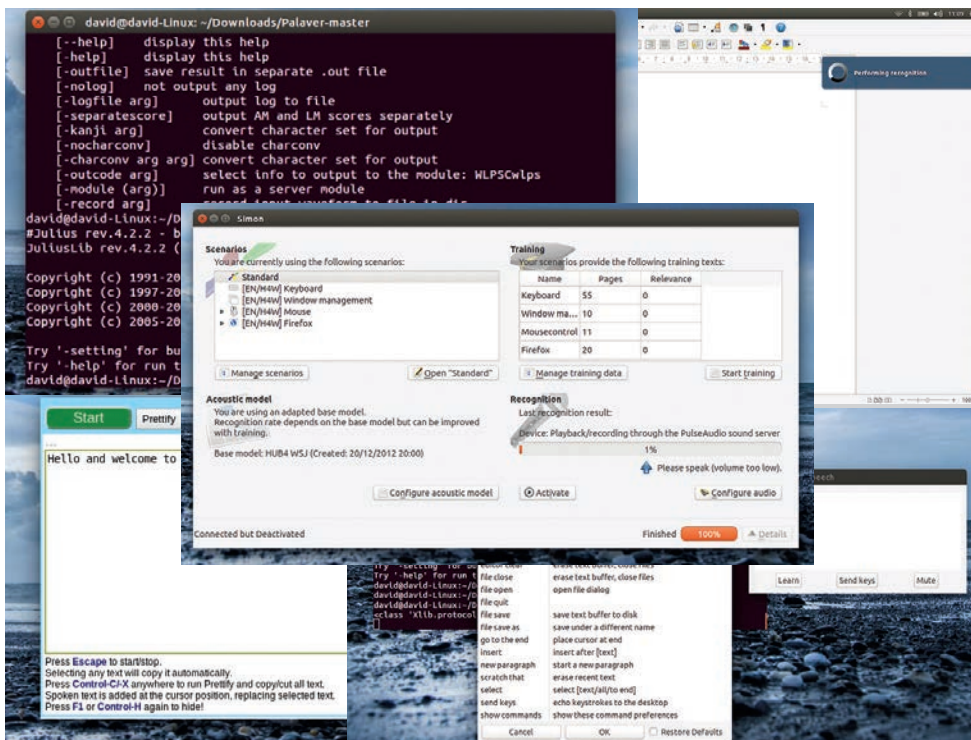
Компания Lenovo на 3-м месте в мире по производству планшетов, 21 квартал подряд лидировала среди фирм-производителей ПК и вошла в четвёрку основных мировых производителей смартфонов, и сейчас продаёт их даже больше, чем ПК; в Китае это продавец смартфонов № 1. LXF

Сравнение

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

Распознавание речи

Дэвид Хейвард нацепил наушники, чтобы протестировать ряд программ распознавания голоса и речи для Linux.



Про наш тест...

Мы решили, что будет разумно взять свежую версию Ubuntu. Почему? Мы сделали допущение (к лучшему или к худшему), что те, кому нужны программы распознавания речи как часть их повседневной жизни, например, слабовидящие студенты, вряд ли располагают временем на возню с разными дистрибутивами, и вместо этого, скорее всего, возьмут самый популярный. Однако мы подметили, что Knoppix (www.knoppix.org) тоже имеет массу спецвозможностей, и есть даже производное Ubuntu специально для людей с плохим зрением, под названием Vinux (<http://vinuxproject.org>).

Мы используем Ubuntu 14.04 64-bit на хорошем стандартном ноутбуке бизнес-уровня, Dell Latitude E6400, с хорошим доступом к портам для наушников и микрофона, 4 ГБ ОЗУ, CPU Intel Core 2 Duo 2,4 ГГц и 80-ГБ жестким диском.

Наша подборка

- » FreeSpeech
- » Julius
- » Palaver
- » Simon
- » Speech-App

Несмотря на героические усилия многих разработчиков, распознавание речи в Linux по большей части значительно отставало от имеющегося на данный момент в других системах. Вот, скажем, серия программ *Dragon Naturally Speaking*. Это превосходный продукт, и, без сомнения, один из лучших программных пакетов распознавания голоса и речи — но предусмотрена ли у него версия специально для Linux? Увы, нет: нам приходится довольствоваться запуском программы через *Wine*.

Раз так, мы решили рассмотреть пять альтернатив Linux, и тут выяснилось, что

ситуация начала меняться, благодаря некоторым выдающимся примерам.

По сути, сдвиги в распознавании речи, которые нам встретились, просто побочный эффект ставшей доступной для публики работы, проводимой Google и прочими, поскольку они продолжают разрабатывать технологию управления голосом для Android, ради обеспечения его конкурентоспособности с Apple. В первом

поколении и Google Now, и Siri от Apple распознавали речь весьма посредственно, но в наши дни эта задача превратилась в повседневную; а как обстоят дела с распознаванием голоса (VR) в разновидностях Linux? Итак, чувствуя себя Скотти в *Звездном пути: Дорога Домой* (правда, мы-то знаем, зачем нужна мышь), мы вступили в беседу с нашим компьютером, как в «Космической одиссее 2001».

«Чувствуя себя Скотти в „Дороге домой“, мы вступили в беседу с нашим компьютером.»

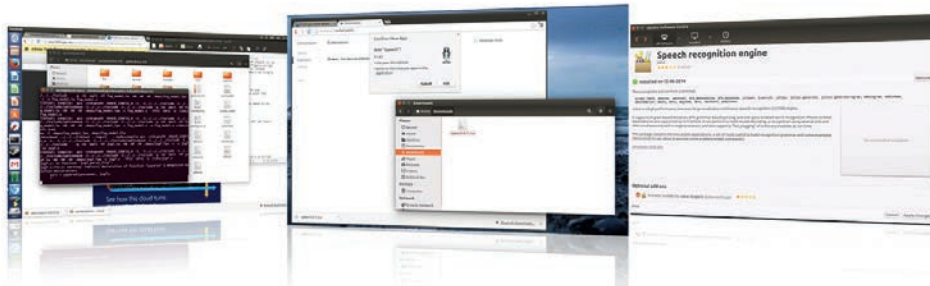
Установка

Легко ли установить подобное приложение?

Во время наших тестов мы обнаружили, что установка этих пакетов распознавания голоса весьма сильно различается. Есть несколько очень простых установок через Software Center, и есть несколько более сложных методов. Конечно, большинство пользователей предпочтут опцию попроще, но, как и для большинства пакетов, более тонкой настройки можно добиться, самостоятельно компилируя исходный код.

Установка *Speech*, или *Speech-App*, замечательно проста. Все, что вам нужно сделать — перейти на страницу Андре Луиса Дос Сантоса [Andre Luiz Dos Santos] в GitHub (<http://bit.ly/speech-app>), затем скачать файл **speech-0.7crx** и перетащить его на страницу Extensions в *Chromium* (считая, что этот браузер уже установлен). Как только расширение туда попадет, оно станет активным, и его можно исполнить через страницу Apps в *Chromium*.

Simon и *Julius* — тоже простые в установке пакеты, поскольку оба расположены в Ubuntu Software Center, и единственное, что от вас требуется — найти их и щелкнуть по кнопке Install. Только учтите: *Simon*



» К счастью, некоторые программы исключительно просты в установке.

не запустится, пока вы не перезагрузите систему.

А вот *FreeSpeech* требует чуть больше усилий, поскольку вам придется предварительно установить и *PocketSphinx*, и *SphinxBase*, а затем — *Pygst* и *Python-lib.display*. Не установив их заранее, мы были вынуждены прибегнуть к своим навыкам Гугл-До, чтобы найти ответы, и спустя 20 минут поиска и попыток расшифровать загадочные объяснения мы обнаружили всякие примочки и применили их.

Palaver [англ. болтовня, суета] вполне оправдал свое название при установке, хотя нам и не пришлось столько суетиться

и блуждать в потемках, как со *FreeSpeech*. Для *Palaver* понадобится исходный код со страницы разработчика *Palaver* Джеймса Мак-Клейна [James MacClain], он же *JamezQ*, на GitHub (<http://bit.ly/Palaver>), а затем установка через терминал *sox*, *Pythonargparse*, *libsox-fmt-mp3*, *mutt*, *xvkbd* и *xautomation*. После этого нам понадобилось запустить файл запуска и настроить горячую клавишу в Custom Shortcuts через Keyboard Settings в Ubuntu System Settings. Однако установка оказалась намного проще благодаря довольно толковому видеоруководству на YouTube от самого Мак-Клейна (<http://bit.ly/PalaverInstall>).

Вердикт

- Julius ★★★★★
- Speech-App ★★★★★
- Simon ★★★★★
- Palaver ★★★★★
- FreeSpeech ★★★★★

» Победитель — *Speech-App*, с быстрым drag-and-drop в *Chromium*.

Простота запуска

Что нужно, чтобы эти голосовые программы начали вас слушать?

Установить все программы распознавания голоса — одно дело, но заставить их нас слушать — совсем другое. И опять здесь блистал *Speech-App*, где всего-то надо было щелкнуть по значку на странице <chrome://apps>,

а затем по зеленой кнопке Start во всплывающем окне.

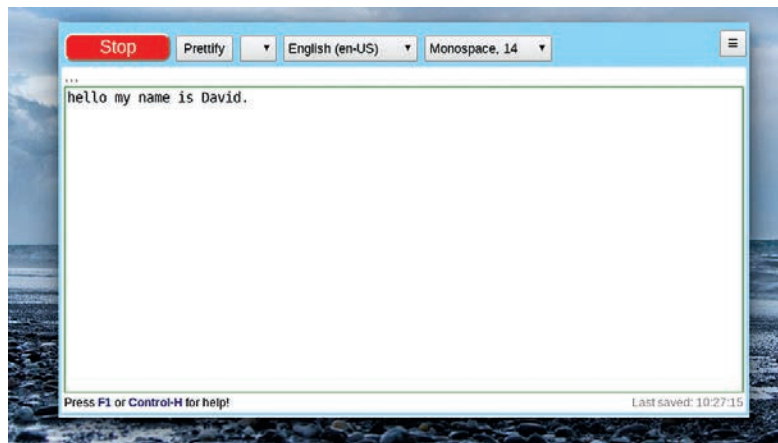
Simon немного сложнее, но это очень умный в запуске инструмент. Появляется мастер запуска, который позволяет настроить ваш голосовой профиль, профиль

сервера, тестовый ввод и вывод звука и настройки громкости.

Запросто устанавливаемый *Julius* за эту услужливость требует подготовки модели целевого языка, которая обрабатывается и запускается через команды терминала и доставляет довольно сильную головную боль при установке и настройке, во многом проводимых методом проб и ошибок.

Freespeech грузится, но активно отказывается слушать перечисленные команды. В общем и целом, нас разочаровали неудобство его использования и отсутствие настройки. Мы не убеждены, что Python-скрипт *Freespeech* работает так, как должен.

Palaver работал вполне прилично, хотя запустить его и заставить слушать было сложнее, чем *Speech-App* или *Simon*. Кроме того, у него была привычка деактивировать горячую клавишу, и когда мы изначально запускали программу и заставляли ее работать, там было немало ошибок.



» *Speech-App* — блестящий пример того, как должно работать распознавание голоса.

Вердикт

- Speech-App ★★★★★
- Simon ★★★★★
- Palaver ★★★★★
- FreeSpeech ★★★★★
- Julius ★★★★★

» *Speech-App* — явный лидер, благодаря умению практически мгновенно начать слушать.

Просторечие и акцент

Ляхко ли с йими? Харашо, панимаэшь?

Мы рассматриваем наши программы распознавания голоса и речи с точки зрения такого пользователя, который не приходит в восторг от слишком длительной возни с первичной настройкой, и в крайнем случае согласен на небольшой адаптационный процесс в начале; но уж затем ему необходимо

нормальное, ровное использование, без излишних проблем и проволочек.

Мы также решили поинтересоваться, хорошо ли каждое из приложений обрабатывает местный акцент, и в данном тесте призвали на помощь двух пользователей. Первый — Джон, его предки родом из Северной Ирландии, и его

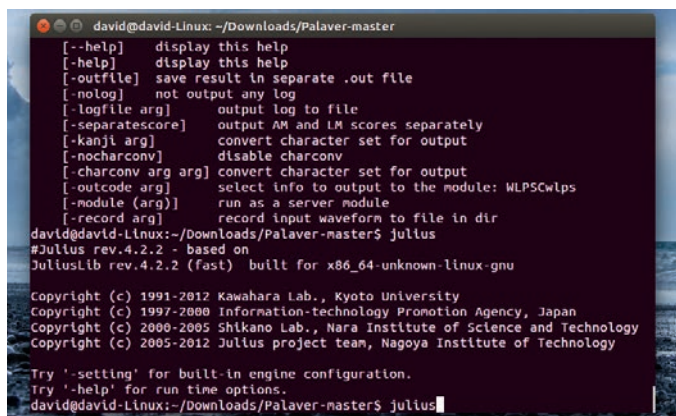
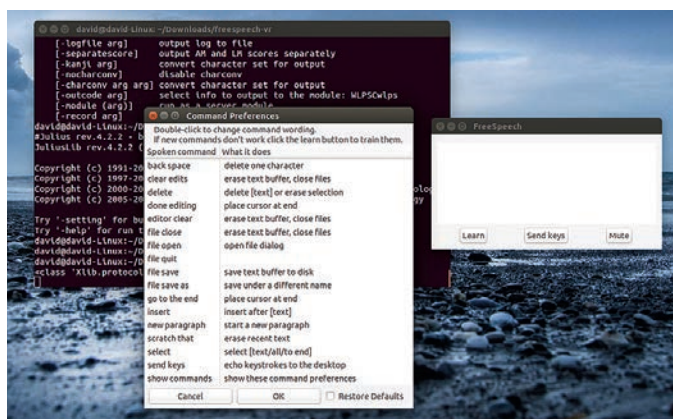
неистребленный акцент буквально режет воздух, как скальпель хирурга. Второй — Мариус, он приехал из России и с нескрываемым удовольствием прожевывает английские слова. Лучших кандидатов для проверки, на что способны наши инструменты распознавания, вряд ли и найдешь. Итак, приступим...

FreeSpeech ★★★★★

С *FreeSpeech* нам не слишком повезло. После начала работы, программа категорически отказалась что-либо делать вообще. Насколько мы поняли, скрипт Python загружается, и графическая сторона интерфейса, например, окно с подробностями команд, вполне живая, но после этого все начинает идти наперекосяк.

К сожалению, мы не в состоянии опубликовать то, что Джон сказал о *FreeSpeech*: мы даже не уверены, что половина этих терминов есть в словаре... и в конце концов, нас ведь и дети читают.

Мариус был несколько сдержаннее нашего ирландского друга, хотя тоже пробормотал пару сочных слов на родном языке, прежде чем сдать ся. И опять-таки мы сильно сомневаемся, что даже переведя эти слова, мы смогли бы их напечатать, а стало быть, больше одной звезды *FreeSpeech* не получит.



Julius ★★★★★

Julius очень быстро стал сучком в глазу. Как мы обнаружили, языковые модели работали не очень хорошо либо вообще не работали. Также оказалось, что команды терминала весьма себе на уме и не делают того, о чем их просят, да и вся программа в целом далеко не так проста в использовании, как следовало бы. Полагаем, что менее опытным пользователям не помешало бы и наличие графического интерфейса.

Не поймите нас превратно: потенциал у приложения явно есть, но на данный момент оно очень ограниченное. Джону повезло — некоторые из произнесенных им слов распознались, но программе не хватило стабильности, чтобы продолжить. С Мариусом вышла такая же ситуация, и в конечном итоге он предпочел *Simon*. Зато нам понравилось, что *Julius* потреблял очень малое количество ресурсов ПК: в общем, менее 32 МБ для среднего рабочего словаря и около 64 МБ для словаря объемом 20 000 слов. Это впечатляет.

Дополнительные функции

Что еще предлагают эти программы?

В *Speech-App* не без пары достойных упоминания функций: прежде всего, можно выбрать язык (из португальского, английского и эсперанто) и изменить шрифт. Можно также «навести красоту» в тексте, добавив пунктуацию, заглавные буквы и специальные символы, например, вопросительный знак и т.д.

В *Simon* функций множество — всех и не перечислишь. Однако главная среди них — опция Server, которая создает центральную базу данных из слов и команд. Это функция, с помощью которой можно

отработать и настроить устную команду или слово. Кроме того, *Simon* может скачать и установить предварительно заданные команды почти для любого элемента рабочего стола Linux и общих программ, включая управление виртуальной клавиатурой.

По сравнению с ним *Julius* бедноват. Но здесь очень нетребовательный к памяти словарь объемом 20 000+ слов — или альтернативный, на 60 000+ — и двухпроходная функция для большей степени точности распознавания.

FreeSpeech использует в качестве внутренней базы данных движок распознавания речи *PocketSphinx*. Как библиотека распознавания, *PocketSphinx* сам по себе весьма уважаем, но слабая реализация *FreeSpeech* уронила его в наших глазах.

Palaver умеет запускать множественный поиск, открывать документы, запускать программы и даже предлагать ответы на незаконченные фразы типа «Англия ведет со счетом», по результатам работы поисковой машины по умолчанию, — вот бы ему еще немного стабильности...

Вердикт

- Simon ★★★★★
- Speech-App ★★★★★
- Julius ★★★★★
- Palaver ★★★★★
- FreeSpeech ★★★★★

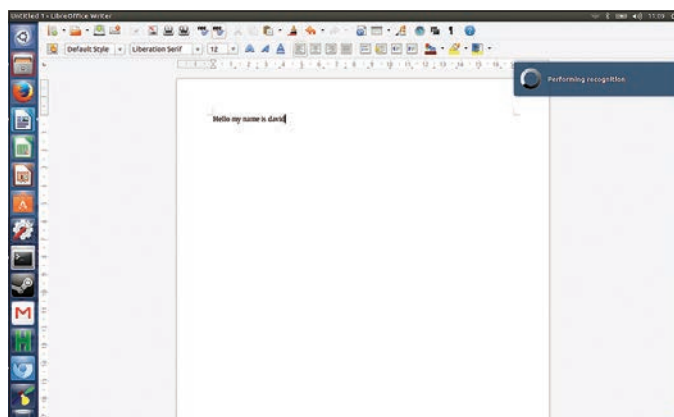
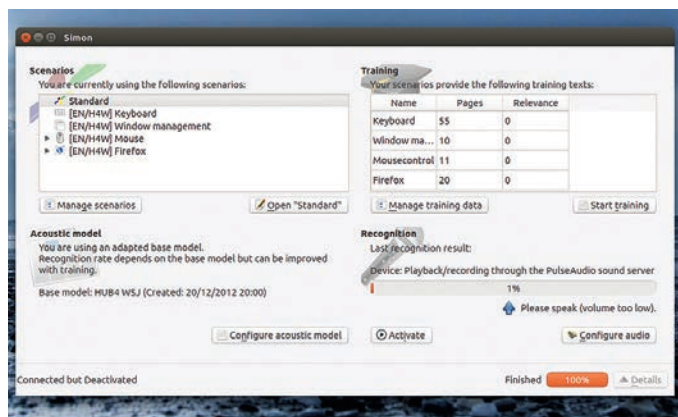
» Simon побеждает благодаря массе дополнительных и удобных функций.

Palaver ★★★★★

Palaver, как мы уже подчеркнули, отличная программа распознавания речи и голоса. На беду, ее подводят серьезные проблемы со стабильностью в работе.

Хотя *Palaver*, как и *Speech-App*, создан на API Google Voice мы обнаружили, что файл горячих клавиш, используемый для привязки выбранной вами клавиши, часто отказывается при постоянном использовании. Вдобавок вы сталкиваетесь с тем, что горячая клавиша или вообще не запускается, или, наоборот, не останавливается ради ввода голосовых команд. Но уж коли она работает, то справляется отлично, и ее так же легко нажимать, как и предвзвешенно заданные клавиши.

И Джон, и Мариус зауважали *Palaver*, когда тот все-таки заработал. Подобно *Speech-App* и *Simon*, он понимал голосовые команды Джона и большинство голосовых команд Мариуса достаточно легко.



Simon ★★★★★

С помощью его умного мастера настройки, *Simon* можно настроить на тихое сидение в фоновом режиме в Linux в ожидании, когда его активируют. В качестве альтернативы, его можно активировать и использовать только когда установлен микрофон. Иными словами, в использовании он до невозможности прост.

Джону опять повезло: все его команды и слова были поняты без проволочек, и он даже отпустил *Simon*'у снисходительный комплимент насчет его простоты в работе, а это — те, кто знает Джона, меня поняли — в его устах высочайшая похвала.

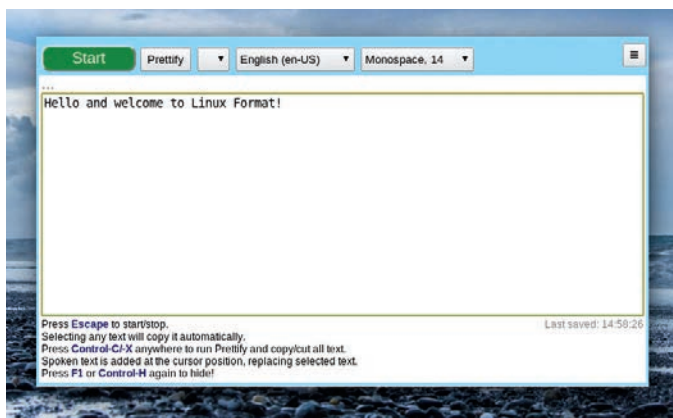
У Мариуса были те же проблемы, что и со *Speech-App* (ниже), но мы смогли настроить *Simon*, чтобы он изучил речевой образец, и даже ввели транскрипцию слов, которые он пытался опознать. Конечным итогом стали почти идеальное распознавание речи и ослепительный русский.

Speech-App ★★★★★

Пока *Speech-App* набирал весьма приличные баллы в наших тестах, и удивляться тут особенно нечему, ведь он использует в качестве внутреннего движка API Google Voice через *Chromium*. Что касается простоты, его достаточно легко установить и запустить, и доступ к нему можно получить в любом порядке и в любой момент, не напрягая рабочий стол, не занимающийся распознаванием голоса.

По части акцента — ирландская речь Джона распознана без особого препирательства. А вот у Мариуса были сложности с распознаванием некоторых слов, и поскольку в *Speech-App* отсутствует функция обучения, улучшить ситуацию было затруднительно.

Тем не менее, с текстовым документом или документом *LibreOffice* (при передаче через окно *Speech-App*) ошибки более или менее поддавались исправлению.



Столкнем лбами

Новички, пользователи среднего уровня и продвинутые — какое им лучше?

Speech-App vs Simon — самый трудный выбор из двух. С одной стороны — простота работы с *Speech-App* для начинающих пользователей и пользователей среднего уровня; его способность мгновенно распознавать голос и речевые образцы и начинать работать; и его простой, но эффективный интерфейс.

С другой стороны, в *Simon* такое тоже есть — плюс еще кое-что. Из-за этого он иногда менее дружелюбен к новичкам. Да, он прост в использовании, но довольно быстро усложняется, причем имеет

привычку оставлять вас в подвешенном состоянии, когда вдруг — по неизвестным причинам — прекращает работать.

Julius vs *FreeSpeech* и *Palaver* — уже не столь затруднительный выбор, и из-за проблем со стабильностью и в *Julius*, и во *FreeSpeech* мы выбираем относительную стабильность, предлагаемую *Palaver*. Когда горячая клавиша *Palaver* работает, программа очень эффективно переводит голосовые команды в действия операционной системы и установленных программ.

Из *FreeSpeech* и *Julius*, мы бы выбрали *Julius*, исключительно благодаря его объемистым словарям. Однако этой паре определенно стоит найти лучший способ реализации и исправить неполадки со стабильностью.

В конечном итоге, все свелось к борьбе между *Speech-App* и *Simon* — где *Simon* обставил *Speech-App* благодаря глубине настройки, лучшему и интуитивному GUI и уровню детализации, которого можно добиться независимо от уровня вашего знания Linux.

Вердикт

- Simon ★★★★★
- Speech-App ★★★★★
- Julius ★★★★★
- Palaver ★★★★★
- FreeSpeech ★★★★★

» *Simon* опять побеждает в рейтинге благодаря большому количеству функций.

Повседневное использование

Как они справляются с повседневной работой?

Одно дело — просто развлекаться с программой распознавания голоса, и совсем другое — использовать ее в качестве повседневного рабочего инструмента или, возможно, для общего управления компьютером. Для тех, кто использует их по медицинским показаниям, это важнейшие технологии доступа, которые обязаны быть надежными.

В свете этого мы полагаем, что *Speech-App* весьма достойно справляется с работой, и считаем, что он предлагает пользователю возможность работать без дополнительного стресса. Кроме того, он прост в установке и настраивается

на работу по многим возможным сценариям. Однако он не идеален, и рано или поздно постоянный пользователь столкнется с его ограничениями.

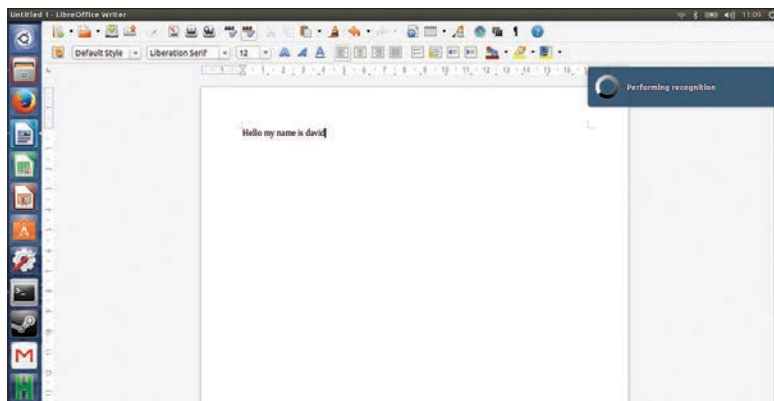
Simon же способен стать отличным дополнением компьютера любого пользователя. Он прост в установке и настраивается под нужды пользователя независимо от среды (дом или офис). И его можно настроить для работы с дополнительными пользователями через функции сервера. Плюс к тому, он относительно прост в поддержке, и в нем есть простор для развития.

Julius, по нашему разумению, в повседневном использовании может вызывать

нешуточные мучения. Наш опыт работы с ним показал, что он немного слишком нестабилен, и из-за нескончаемых проблем доводит пользователей до лишнего стресса. Мы полагаем, что конечный пользователь вряд ли его оценит и, скорее всего, предпочтет нечто более управляемое — например, *Simon*.

FreeSpeech ничуть не лучше, если не хуже. Его можно превратить в рабочее решение, но затраченные на это время и усилия отпугнут от него большинство пользователей и заставят их обратиться к чему-нибудь еще, тем более при наличии лучших программ.

Palaver способен прекрасно справиться с работой, при условии, что все, что должен делать пользователь — это инициировать горячую клавишу для запуска процесса; в этом случае все будет хорошо. Но в настоящее время после каждой голосовой команды горячую клавишу приходится вводить снова, чтобы она обрабатывалась и выполнялась программой. Одно это уже делает ситуацию не идеальной даже для эпизодического, разового пользователя, не говоря о тех, кому приходится пользоваться им постоянно. На исправление ошибок и необходимость перезапускать процесс при отказах программы — не в меру частых — уходит слишком много времени.



» Когда *Palaver* работает, это хороший инструмент распознавания голоса. Однако в нем слишком много источников проблем — и нередких.

Вердикт

Simon ★★★★★

Speech-App ★★★★★

Palaver ★★★★★

FreeSpeech ★★☆☆☆

Julius ★★☆☆☆

» Как надежная программа для повседневного использования, призывает Simon.

Дальнейшее развитие

Что ждет в будущем эти программы распознавания речи?

Похоже, что программы распознавания речи в Linux не могут похвастаться такой же движущей силой, как на других платформах. *Speech-App* в последний раз обновлялся четыре месяца назад (на момент написания), и все еще находится в стадии разработки; надемся, вскоре он обретет больше функций.

Simon выказал активность разработки примерно с той же давностью, запустив в апреле этого года версию для Mac OS X и как минимум продемонстрировав, что команда по-прежнему занимается своей программой и улучшает ее.

Julius последние полгода в затишье; последнее обновление вышло в середине января. С оглядкой на историю его обновлений, через несколько месяцев мы, вероятно, все-таки увидим новую версию.

Ситуация с *FreeSpeech* прямо противоположная. Последний раз его пакет кода Google обновлялся в октябре 2013 года, и хотя до сих пор есть пользователи, которые продолжают с ним работать, ему отчаянно требуется, чтобы кто-нибудь как следует перешерстил скрипты Python и процедуры установки.

Последнее обновление *Palaver* было в январе, а сейчас появилось предупреждение, что «с 1 января 2014 года поддержка *Palaver*, предоставленная Бреттом Мейсоном [Brett Mayson], прекращается... Сейчас я занят проектом *LiSpeak* — (<http://lispeak.bmandesigns.com>), и он



» *Simon* слушает... и адаптируется к будущему в распознавании голоса в Linux.

основательно пересмотрен ради лучшей поддержки большего количества языков и упрощения разработки плагинов». Мы связались с Мейсоном, и он сказал нам, что из-за изменений в API Google разработка *LiSpeak* замедлилась, но команда надеется, что к концу августа он «будет работать лучше, чем когда бы то ни было».

Вердикт

Simon ★★★★★

Speech-App ★★★★★

Julius ★★☆☆☆

FreeSpeech ★★☆☆☆

Palaver ★★☆☆☆

» И *Simon*, и *Speech-App* недавно обновлялись, и мы считаем это ничьей.

Распознавание голоса

Вердикт

Как мы убедились, распознавание речи в Linux может быть вполне на уровне популярных программ вроде *Dragon Naturally Speaking*. Однако заставляет задуматься: что случилось бы, если бы команда разработки *Simon* смогла уделить чуть больше времени разработке, или проект *Speech-App* получил бы вдруг толчок в нужном направлении? Тогда мы бы стали свидетелями того, как другие платформы смотрят на программы распознавания речи в Linux с завистью, и, возможно, даже того, что они станут фактическим стандартом.

Выбор программы распознавания голоса и речи **LXF** свелся к схватке между *Speech-App* и *Simon*: оба предлагают пользователю лучшую систему распознавания, чем трое остальных.

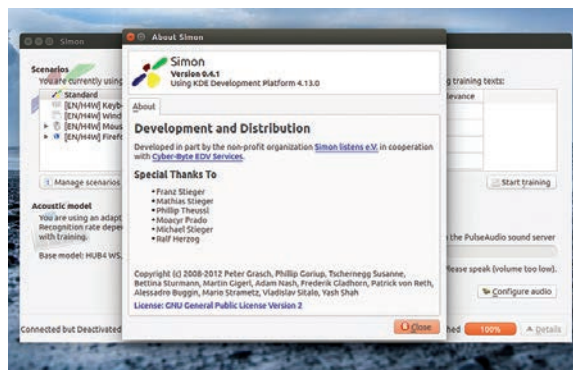
В данном Сравнении *Simon* становится победителем, хотя *Speech-App* уступает с очень небольшим отрывом. *Simon* —

более полная и всесторонняя программа, которая, без сомнения, станет конкурентом от Linux таким программам, как *Dragon Naturally Speaking*.

Julius занимает третье место, и над ним стоило бы поработать, в частности, чтобы сделать его дружелюбнее к новичкам.

Palaver уступает *Julius* лишь слегка, но, поскольку он больше не разрабатывается, и его создатель, Бретт Мейсон, перешел на *LiSpeak*, вероятно, не стоит тратить время на его изучение, если вас интересует долгосрочная перспектива своей программы распознавания речи в Linux. Мы с Мейсоном на связи, и непременно рассмотрим основанный на *Palaver LiSpeak*, когда команда будет готова продемонстрировать результаты своей работы.

И, наконец, у нас есть *FreeSpeech*, с которым мы так и не справились по причине его



» *Simon* говорит [популярная игра вроде «испорченного телефона», — прим. пер.]... Точнее, *Simon* делает то, что ему говорят.

капризной установки и скриптов Python, склонных весьма зрелищно выдавать отказ. Однако это дает шанс читателям **LXF** со способностями к Python протянуть руку помощи.

«Simon становится победителем — как всесторонняя программа.»

I Simon ★★★★★
 Сайт: <http://simon.kde.org> Лицензия: GNU GPL v1.2 Версия: 0.4.1
 » «Simon Слушает» — такой у него слоган. И мы можем подтвердить, что это правда.

IV Palaver ★★★★★
 Сайт: <http://bit.ly/Palaver> Лицензия: GPL v3 Версия: 1
 » Разработка Palaver прекратилась. Сумеет ли LiSpeak занять его место?

II Speech-App ★★★★★
 Сайт: <http://bit.ly/speech-app> Лицензия: GNU GPL Версия: 0.5
 » Простота и дружелюбный подход Speech-App делают его достойной программой.

V FreeSpeech ★★★★★
 Сайт: <http://bit.ly/Freespeech> Лицензия: GPL v3 Версия: r120
 » Над скриптами Python во FreeSpeech надо поработать — может, возьметесь?

III Julius ★★★★★
 Сайт: <http://bit.ly/JuliusEng> Лицензия: <http://bit.ly/JuliusLic> Версия: 4.2.2
 » Став человечнее, Julius способен оказаться блестящим инструментом.

Обратная связь
 Нашли пакет распознавания голоса в Linux лучше наших? А может, вы как раз над таким работаете? Сообщите нам свои мнения: lxfl.letters@futurenet.co.uk.

Рассмотрите также...

Мы не стали бы отказываться от специальных программ Linux чисто по соображениям простоты, но иногда это стоит сделать, хоть мы и предпочли бы видеть родные альтернативы Linux и альтернативы с открытым кодом.

Распознавание речи в Linux — именно такой случай: отличные рабочие результаты дает применение программы Windows типа *Dragon Naturally*

Speaking. Мы протестировали версию 11.0 в *Wine* и можем подтвердить, что она работает хорошо. Была пара проблем с отключением микрофона, но когда мы его быстро выдернули и воткнули опять, все заработало.

Помимо запуска программ в *Wine*, на горизонте появился интересный проект под названием *SpeechControl* для Ubuntu, более подробную информацию о котором вы сможете обнаружить

на <http://bit.ly/SpeechControl>. Если все пойдет соответственно плану разработчика, в *SpeechControl* будут такие функции, как Text to Speech, Speech Recognition, Virtual Assistant, плюс набор предварительно настроенных программ, исполняемых в любом дистрибутиве Ubuntu или на базе Ubuntu. В конечном итоге из *SpeechControl* должно получиться мощное дополнение к репозиториям дистрибутива. **LXF**

РАЗГОН LINUX

Засучите рукава и ступайте в гараж за Маянком Шармой, чтобы организовать вашему компьютеру Linux высокооктановый разгон.



По сравнению с другими популярными проприетарными операционными системами, типичный настольный дистрибутив Linux по умолчанию уже представляет собой отлично смазанную машину. На каждом этапе сборочной линии дистрибутива Linux, от ядра до индивидуальных приложений, разработчики вкладывают титанические усилия, обеспечивая оптимальное использование ресурсов оборудования разными программами. Именно по этой причине вы можете использовать — и весьма продуктивно — Linux на старых одноядерных процессорах с меньшим объемом ОЗУ (Random Access Memory, RAM), чем на современном смартфоне.

Но нет предела совершенству! Разработчики дистрибутивов Linux адресуются к широкой

аудитории, и им приходится обеспечивать совместимость с разным оборудованием. А значит, ваш дистрибутив, возможно, работает с настройками, не оптимизированными именно под вашу машину. Вдобавок многие современные дистрибутивы набиты программами и сервисами, из-за которых

«Типичный дистрибутив Linux весьма гибок и допускает настройку на ваше усмотрение.»

прямо спят на ходу. Но хорошая новость в том, что типичный дистрибутив Linux весьма гибок и допускает настройку на ваше усмотрение, например, добавление или удаление некоторых программ.

Подобным же образом, чтобы выжать из своего дистрибутива максимум производительности,

придется залезть под капот и настроить ряд важных компонентов, избавившись от излишеств. Настроить компьютер с Linux для ускорения работы не особо сложно, благодаря открытой природе кода работающих в нем программ. Однако этот процесс может осложниться, и потребовать знания и знакомства с основными компонентами вашего дистрибутива.

В данной статье мы проведем вас по всему процессу превращения вашей рабочей лошади в скакового жеребца, независимо от вашего уровня подготовки. Более того, эти советы пригодятся как на современных многоядерных машинах, так и на старых одноядерных, не слишком богатых ресурсами, и мы научим вас проскакивать узкие места в производительности.

Хитрости настольной работы

Ускорьте загрузку, работу и отклик приложений.

Быстрые компьютеры любят все. В данном разделе мы рассмотрим основные приемы, способные ускорить работу именно вашего компьютера. Чтобы извлечь максимум из своего компьютера с Linux, не обязательно быть опытным пользователем. Существуют методы заставить дистрибутив Linux грузиться быстрее, полезные даже новичкам.

Двойная ли у вас загрузка Linux с другой ОС или нет, после установки вашего любимого дистрибутива Linux в процесс загрузки однозначно вмешается загрузчик *Grub*. По умолчанию, большинство дистрибутивов Linux отобразят *Grub* откуда угодно за 10–30 секунд.

Один из простейших способов заставить ваш компьютер грузиться быстрее — подстричь продолжительность работы загрузчика. Если вы всегда выбираете опцию по умолчанию и любите приключения, можете даже вовсе пропустить обратный отсчет, хотя мы бы не рекомендовали этого делать.

Чтобы подправить обратный отсчет *Grub*, запустите терминал и откройте файл `/etc/default/grub` в текстовом редакторе, например,

```
sudo nano /etc/default/grub
```

— здесь выбран редактор *Nano*. Выловите в файле переменную `GRUB_TIMEOUT`: она задает длительность показа загрузчика. Затем замените значение данной переменной на нечто вроде 3 или 5 (это длительность в секундах). Установка ее в 0 отменит обратный отсчет, и *Grub* будет сразу загружать ОС по умолчанию. Установив новый таймер обратного отсчета, сохраните файл и сообщите *Grub* о новых настройках командой `update-grub`.

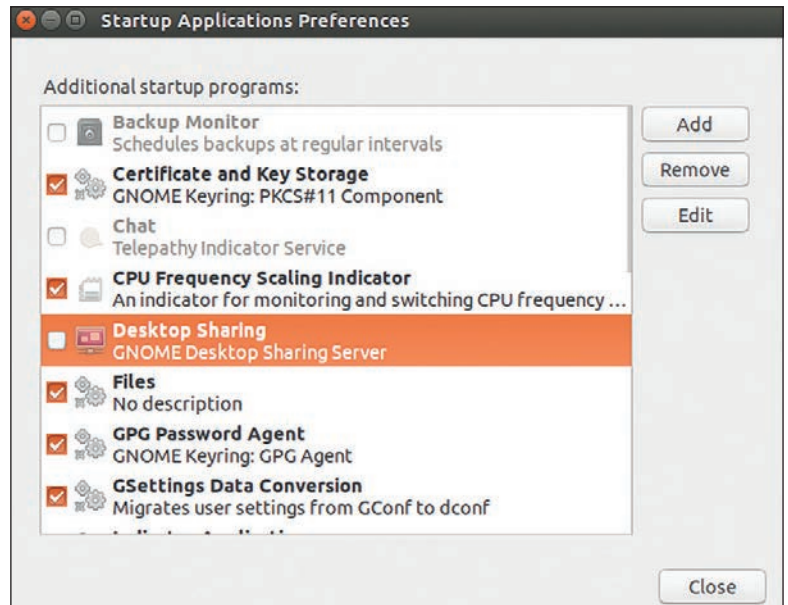
Ускорьте сервисы загрузки

Одна из основных причин долгой загрузки в том, что ваша система запускает избыточные приложения и сервисы. Но прежде чем начать от них избавляться, неплохо составить себе представление о том, что происходит, пока ваш дистрибутив загружается.

Создать профиль процесса загрузки Linux и измерить время загрузки разных сервисов поможет несложное приложение *Bootchart*. Его вы можете использовать для определения проблемных мест в процессе загрузки. *Bootchart* регистрирует любую деятельность, а затем выводит результаты в подробный файл образа.

Вы найдете *Bootchart* в репозиториях всех основных дистрибутивов, в том числе Ubuntu, Fedora, openSUSE и Mageia. Используйте менеджер пакетов своего дистрибутива, чтобы найти и установить пакет *bootchart*. Установив приложение, перезагрузите компьютер. Когда появится рабочий стол, перейдите в папку `/var/log/bootchart`. Здесь вы найдете файл образа (с расширением `.png`), который содержит анализ последнего процесса загрузки. При каждой последующей загрузке будет создаваться новый файл, с отметкой времени.

Вверху перечислены всякие данные статистики: дата теста, название дистрибутива, версия ядра и включенные опции ядра, а также время, затраченное на загрузку системы. За ней следуют две диаграммы, показывающие загрузку CPU и активность диска на стадии загрузки, а затем идет основная часть анализа. Этот основной раздел содержит ряд маркированных полос, каждая из которых представляет собой отдельный процесс. Некоторые полосы обрываются через пару секунд. Те, что работают до самого конца, обозначают сервисы, среди прочих — такие, как *Network Manager*, *Cron* и демон CUPS. Образ также показывает дочерние



процессы, соединяя их с родительскими процессами пунктирными линиями.

По данному образцу вы можете узнать, какие процессы активны, и удалить те, которые вам ни к чему. Например, если вы печатаете только изредка, можно отключить запуск CUPS при загрузке. Более того, этот образ также поможет вам определить процессы, которые берут на себя управление всеми ресурсами и заставляют другие процессы ждать, по сути блокируя процесс загрузки.

Теперь, когда вы понимаете, как загружается ваш компьютер и как запускаются сервисы, пора подогнать процесс загрузки под ваши потребности и высвободить некоторое время. Простейший способ высвободить ресурсы в любом дистрибутиве

» Изменив файлы отдельных приложений в `/etc/xdg/autostart`, можно увидеть скрытые по умолчанию приложения в *Startup Applications*.

«Простейший способ высвободить ресурсы — остановить ненужные процессы.»

Linux — остановить работу ненужных процессов или вообще отменить их загрузку. В большинстве дистрибутивов Linux есть инструмент, позволяющий увидеть, что происходит, и при необходимости остановить это.

Ubuntu поставляется с инструментом *Startup Applications*, через который добавляются и удаляются любые приложения, которые вы бы хотели запускать при загрузке. Для более быстрой загрузки, запустите это приложение и отключите все ненужные приложения, которые там обнаружатся.

По умолчанию *Startup Applications* не показывает все приложения и сервисы. Чтобы увидеть скрытые сервисы и приложения, не имеющие GUI, откройте терминал и с помощью `cd /etc/xdg/autostart` перейдите в директорию, где перечислены все сервисы. Там вы найдете по файлу на каждое установленное приложение и сервис. Во всех файлах есть переменная, контролирующая,

»



показан ли сервис в инструменте *Startup Applications* или нет. Вы можете изменить значение переменной по умолчанию в каждом файле с помощью

```
sudo sed --in-place 's/NoDisplay=true/NoDisplay=false/g'
*.desktop
```

Перезапустив инструмент *Startup Applications*, вы найдете дополнительные программы запуска, например, *Desktop Sharing*, *Personal File Sharing*, и т.д. Прочтите их описания и отключите те, что вам не нужны. Чтобы отключить загрузку сервиса, выберите сервис и просто уберите галочку из окошка рядом с его названием. Только не нажимайте на кнопку *Remove*, и тогда впоследствии вы сможете при желании снова его включить. Также не стоит отключать пункт автозапуска, если вы не вполне понимаете, что он делает, в противном случае это может привести к обратному

здесь на вкладку *Preview* и установите значение *Show thumbnails* в *Never*.

Кроме того, пользователи Ubuntu должны установить *Compiz-Config Settings Manager*, с помощью

```
sudo apt-get install compizconfig-settings-manager
compiz-plugins-extra
```

чтобы получить возможность изменять эффекты рабочего стола. Пользователям Gnome надо аналогично установить инструмент *Gnome Tweak Tool*. Он имеется в официальных репозиториях основных настольных дистрибутивов на базе Gnome, таких, как Fedora. Пользователи, применяющие RPM, могут установить его с помощью `sudo yum install gnome-tweak-tool`, а те, кто применяет Deb — с помощью `sudo apt-get install gnome-tweak-tool`. Если вы используете KDE, перейдите в *System Settings*, найдите эффекты рабочего стола и отключите их.

Подобным же образом, функции *Nepomuk*, *Strigi* и *Akonadi* на рабочем столе KDE потребляют ресурсы памяти. *Nepomuk* и *Strigi* вы можете отключить через *System Settings*, перейдя в раздел *Desktop Search*. Чтобы отключить *Akonadi*, выключите сервер *Akonadi* командой `sudo akonadictl stop`. Затем откройте файл `/etc/akonadi/akonadiserverrc` в текстовом редакторе и измените параметр *StartServer* с *True* на *False*.

Поскольку *Akonadi* тесно связан с рабочим столом KDE, то при запуске какого-либо приложения, завязанного на *Akonadi*, будет автоматически запускаться сервер *Akonadi*. Некоторые исполняемые файлы *KRunner* и виджеты *Plasma* также используют *Akonadi*, так что вам придется отключить и их тоже. Чтобы отключить работающие с *Akonadi* исполнители *KRunner*, нажмите *Alt+F2* и щелкните по значку с гаечным ключом. Затем снимите галочку с исполнителей *Nepomuk Desktop Search* и *Instant Messaging Contacts*. Далее надо сообщить виджету *Digital clock*, чтобы он не отображал календарных событий — сделайте это, щелкнув правой кнопкой по цифровым часам на панели, а затем перейдя в *Digital Clock Settings*. Перейдите во вкладку *Calendar* и снимите галочку с опции *Display Events*.

Приложение индексации *apt-xapian-index* ускоряет часть операций поиска, но способно оказать сильное негативное воздействие на производительность более слабых компьютеров. Можете смело удалять этот пакет с помощью `sudo apt-get purge ptxapian-index`, поскольку он не является жизненно необходимым.

«Интересные графические эффекты не подходят для машин с ограниченными ресурсами.»

эффекту по удобству использования. Скажем, если в вашем компьютере нет оборудования Bluetooth, спокойно отключайте апплет *Bluetooth Manager*. А вот если вы отключите *Mount Helper*, Ubuntu прекратит автоматическое монтирование подключаемых устройств.

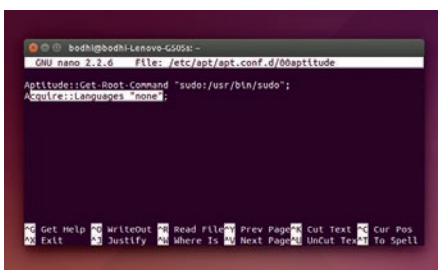
Отключите украшательство

Современные рабочие столы идут с интересными эффектами, чтобы несколько оживить обычные задачи, например, открытие и закрытие окон и приложений. Однако графические эффекты не слишком подходят для машин с ограниченными ресурсами, и поэтому должны быть немедленно отключены.

Некоторые интересные функции мы принимаем как должные — скажем, предпросмотр эскизов в менеджере файлов. Если просматривать папку с парой файлов, это нормально. Но откройте папку с сотнями файлов на медленной машине, и менеджер файлов начнет расточать драгоценные ресурсы на создание эскизов.

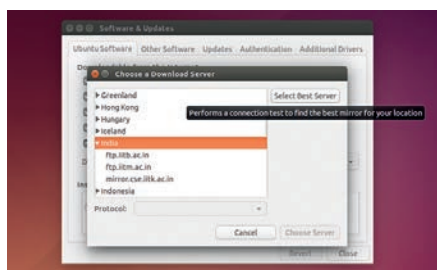
Чтобы отключить эскизы на машине с Gnome, запустите менеджер файлов и перейдите в *Edit > Preferences*. Переключитесь

Устанавливаем приложения быстрее



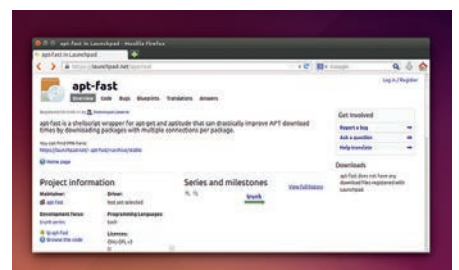
1 Урежем язык

Результатом команды `sudo apt-get update` будут три типа строк — *hit*, *ign* и *get*. Большинство строк *ign* относятся к языковым переводам. Если все приложения у вас на английском, можно ускорить обновления, убрав те, что касаются языков. Для этого добавьте `Acquire::Languages "none"`; в файл `/etc/apt/apt.conf.d/00aptitude`.



2 Возьмем ближайшее зеркало

Если ваш дистрибутив — Fedora, установите с помощью `sudo yum install yum-plugin-fastestmirror` плагин *fastestmirror*: он рассчитает время соединения со всеми доступными зеркалами и выберет самое быстрое из них. Пользователи Ubuntu должны загрузить приложение *Software & Updates*, выбрать опцию *Other* во всплывающем меню *Download From* и нажать на кнопку *Select Best Server*.



3 Применим apt-fast

Apt-fast — замена *apt-get*, которая ускоряет обновления, скачивая пакеты при множественном соединении. Добавьте PPA *apt-fast* с помощью `sudo add-apt-repository ppa:apt-fast/stable` и установите этот пакет с помощью `sudo apt-get install apt-fast`. Для самого последнего релиза Ubuntu скрипта нет, и вам придется добывать пакеты вручную.

Если вы работаете с дистрибутивом на ноутбуке, нужно кое-что принять во внимание. Ваш среднестатистический ноутбук — это, по сути, как бы две машины. Пожелав протестировать его производительность, вы бы в одном и том же дистрибутиве получили разные результаты. Причина в том, что производительность ноутбуков сама настраивается в зависимости от энергопотребления. Работая на батареях, ноутбук пытается экономично расходовать их заряд, и поэтому снижает свою производительность.

Вы можете получить больший контроль над этим процессом благодаря инструменту *TLP* — это продвинутый инструмент командной строки для управления энергопотреблением в Linux, который пытается сохранить заряд батарей, одновременно максимизируя производительность.

В Ubuntu можно установить *TLP*, сначала добавив PPA с помощью `sudo add-apt-repository ppa:linrunner/tlp`, а затем обновив репозитории с помощью `sudo apt-get update`; сам инструмент устанавливается с помощью `sudo apt-get install tlp tlp-rdw`. Пользователям Fedora надо ввести следующие три команды:

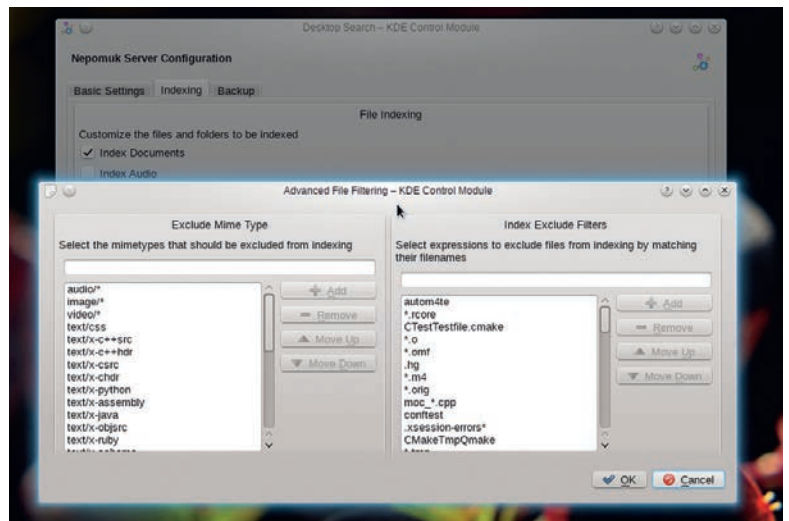
```
sudo yum localinstall --nogpgcheck http://repo.linrunner.de/fedora/tlp/repos/releases/tlp-release-1.0-0.noarch.rpm
sudo yum localinstall --nogpgcheck http://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-stable.noarch.rpm
sudo yum install tlp tlp-rdw
```

После установки запустите *TLP* командой `sudo tlp start`. Хотя он работает в фоне, ряд настроек надо применить вручную, чтобы заменить параметры по умолчанию *TLP*, такие, как включение или выключение по требованию радио Wi-Fi и Bluetooth и переключение между настройками AC или батарей невзирая на реально выбранный источник питания. Эти настройки можно изменить, отредактировав файл настройки *TLP*, который находится в `/etc/default/tlp`.

Далее, пользователи Ubuntu, желающие контролировать настройку мощности CPU своего ноутбука, должны также установить индикатор *CPUFreq*, с помощью `sudo apt-get install indicatorcpufreq`. После его установки можно будет контролировать функцию настройки CPU через индикатор на верхней панели меню.

Проверьте свой жесткий диск

Обновили вы свой компьютер, добавили пару гигабайт ОЗУ и многоядерный процессор, но... почему-то не ощутили особой разницы в скорости загрузки и запуска приложений. Возможно, причина в том, что вы не позаботились обновить оборудование, которое



» Отключать индексацию файлов KDE не обязательно: ее можно индивидуально настроить через параметры.

берет на себя немало тяжелой работы: старый добрый жесткий диск. Большинство дистрибутивов включают инструмент командной строки *hdparm*, пригодный для проверки и тестирования жесткого диска. Считая, что ваш диск монтирован на `/dev/sda`, используйте `sudo hdparm -I /dev/sda` для получения исчерпывающей информации о своем диске, в том числе о скорости вращения его «блинов», которая для потребительских жестких дисков обычно между 5400 и 7200 об./мин. Чем быстрее вращение, тем выше скорость передачи.

Команда `sudo hdparm -t --direct /dev/sda` измерит скорость чтения данных диска в МБ/с. Чтобы измерить скорость записи, создайте файл с помощью команды `dd`, например, `dd if=/dev/zero of=readtest bs=8k count=500000`.

Отобразится время на создание этого файла вместе со средней скоростью записи. Не забудьте потом удалить файл `readtest`, он ест 4 ГБ.

Эти два параметра, вместе с другими, такими, как фрагментация системы, сильно влияют на скорость компьютера. Обновляя свой компьютер, помните, что типичная скорость чтения/записи традиционного жесткого диска составляет 120 МБ/с, а скорость SSD (Solid State Drive) обычно доходит до 200 МБ/с.



Избавьтесь от мусора

Со временем обычный дистрибутив Linux создает целое хранилище ненужных данных. Эти забытые файлы и артефакты данных могут таинственным образом замедлять работу компьютера. Вот почему, помимо использования упомянутых в этой статье функций, вы должны регулярно уделять время на наведение порядка в своем дистрибутиве.

Ваш дистрибутив создает и хранит ярлыки в скрытых директориях даже после удаления исходного файла. Со временем количество этих ярлычков увеличивается до невероятных размеров. Перейдите в скрытую директорию `.cache/thumbnails/` и удалите файлы во всех поддиректориях.

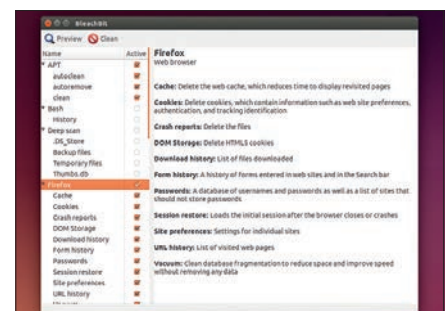
Если вы усердно обновляете свой дистрибутив, есть вероятность, что у вас скопилось немало старых ядер, которыми вы уже не пользуетесь. Они сохраняются на тот случай, если вы не сможете загрузить свой дистрибутив с обновленным ядром,

и обычно перечислены в разделе Advanced options в меню *Grub*.

Чтобы удалить неиспользуемые ядра, сначала найдите версию, которой вы пользуетесь на данный момент, по команде `uname -r`. Затем используйте менеджер пакетов своего дистрибутива, чтобы найти все пакеты, начинающиеся с `"linux-image"`, и удалить не соответствующие текущей версии ядра. Пользователи Fedora могут применить команду `sudo yum remove kernel` — она автоматически сотрет все неиспользуемые ядра; а пользователи Ubuntu помогут инструмент *Ubuntu Tweak*.

Стоит также почистить кэш системы управления пакетами. Команда `sudo yum clean all` наведет порядок в *Yum* для пользователей Fedora, а пользователи Ubuntu должны использовать команду `sudo apt-get autoremove` для удаления забытых пакетов, а затем `sudo apt-get autoclean`, чтобы удалить

частично установленные пакеты или пакеты, которые больше не установлены в системе.



» Установите через менеджер пакетов инструменты типа *BleachBit*, чтобы ваш дистрибутив был в порядке.

Продвинутые приемы

Ужмите раздел подкачки и индексации и почистите ядро.

Описанные на предыдущих страницах приемы позволят вам извлечь максимум из своего дистрибутива Linux, приложив минимальные усилия. А если вы все равно не утолили свой аппетит к скорости, давайте перейдем к более продвинутой технике. Однако прежде чем применять эти приемы, помните, что они не только сложнее: при небрежном применении они могут вызвать серьезные последствия для вашей системы.

Давайте начнем с RAM-диска — размещенной в ОЗУ файловой системе, которая создает область хранения прямо в ОЗУ вашего компьютера, как будто это раздел на жестком диске. Основное преимущество здесь в том, что RAM-диски очень быстрые. Поскольку данные на них теряются при перезагрузке, RAM-диски подходят только тем приложениям, которым постоянно нужны небольшие области данных для кэширования или для использования в качестве временного хранилища.

Перед созданием RAM-диска скомандуйте **free**, чтобы выяснить объем неиспользованного ОЗУ в вашем дистрибутиве. Затем создайте папку, чтобы употреблять ее как точку монтирования вашего RAM-диска, с помощью **sudo mkdir /mnt/ramdisk**, и смонтируйте его с помощью

```
sudo mount -t tmpfs -o size=1024m tmpfs /mnt/ramdisk.
```

Создастся RAM-диск на 1 Гб с файловой системой tmpfs, разработанной специально для таких дисков.

Чтобы при каждой загрузке дистрибутива RAM-диск монтировался автоматически, вам придется добавить его в файл **/etc/fstab**. Откройте его в текстовом редакторе и введите следующее в новой строке:

```
tmpfs /mnt/ramdisk tmpfs nodev,nosuid,noexec,nodiratime,size=2048M 0 0
```

Создав RAM-диск, вы можете разместить на нем кэш своего браузера. Если у вас *Chromium*, то все, что вам нужно сделать — это изменить пункт программы загрузки, чтобы указывать на RAM-диск соответствующим переключателем, например, **--disk-cachedir="/mnt/ramdisk"**.

Если вы используете *Firefox*, введите **about:config** в адресную строку. Щелкните правой кнопкой где-нибудь на списке и выберите **New > String**. Когда у вас спросят имя параметра, введите **browser.cache.disk.parent_directory**, а в качестве его значения введите путь, где вы хотите хранить кэш, например, **/mnt/ramdisk**. Далее, найдите параметр **browser.cache.disk.enable** и убедитесь, что он настроен на **True**; если это не так, щелкните по нему дважды, чтобы переключить его.

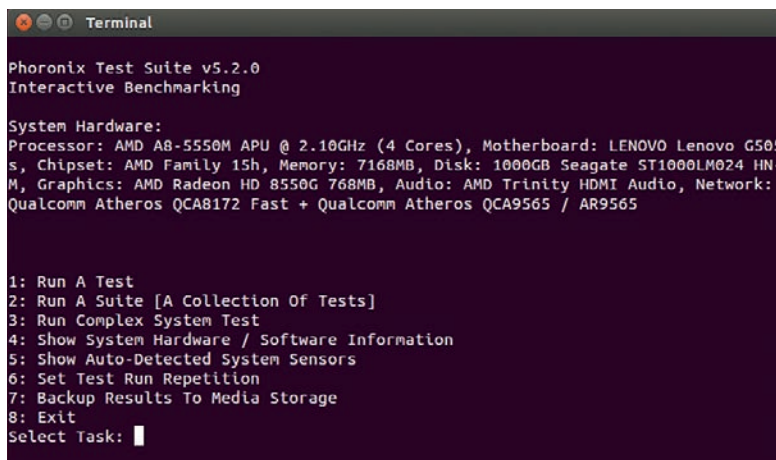
Контролируйте подкачку

Вам невдомек, почему ваша обремененная огромным ОЗУ машина по-прежнему тормозит? Весьма вероятная причина — то, что ваш дистрибутив по-прежнему продолжает использовать пространство подкачки [swap] на жестком диске, отчего скорость чтения/записи меньше, чем в ОЗУ.

Если ОЗУ в вашей системе хватает, чтобы справиться со всем, что вы ей ни швырнете, минимизируйте использование раздела подкачки. Параметр подкачки [swappiness] управляет способностью ядра перемещать процессы из физической памяти на swap-диск, что может увеличить время отклика приложения, если процессы выпихиваются из памяти.

Команда **cat /proc/sys/vm/swappiness** отобразит текущий уровень swappiness, заданный вашим дистрибутивом. В Ubuntu это значение равно 60. Данный параметр принимает значения между

► Подумайте о применении инструмента тестирования, например, **Phoronix Test Suite**, для регулярного контроля производительности своего компьютера.



Анализируйте и оптимизируйте загрузку

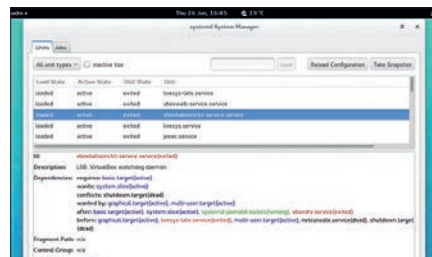
Мы уже рассматривали графические инструменты, имеющиеся в каждом дистрибутиве, которые помогают избавиться от ненужных приложений и сервисов. Теперь рассмотрим, как дальше оптимизировать загрузку через *Systemd* (см. Учебники, стр. 62). Это демон управления системой, созданный для замены стареющей системы *init*. Подобно *init*, *Systemd* является главным демоном, управляющим другими демонами. Основные настольные дистрибутивы, такие, как Fedora, Mageia и openSUSE, уже перешли на *Systemd*, а остальные, например, Debian и Ubuntu, сообщили о своих планах перехода в грядущих релизах.

В дистрибутиве, использующем *Systemd*, можно измерить время загрузки командой **systemd-**

analyze, которая распечатает время загрузки ядра, сервисов и пользовательской среды. **systemd-analyze blame | head** выдаст вам список процессов, занимающих большую часть времени загрузки, упорядоченный по времени инициализации. Команда **systemd-analyze plot > boot.svg** визуализирует весь процесс загрузки в файле изображения **boot.svg**. Вы можете посмотреть этот файл в браузере, чтобы получить представление о потоках процессов.

Определив проблемный сервис, вы можете его замаскировать. Тогда его нельзя будет запустить с помощью *Systemd*, и он по сути станет невидимым. Если вы не используете LVM, RAID или зашифрованные устройства, можете замаскировать практически все сервисы хранения Fedora, такие,

как **sudo systemctl mask lvm2-monitor.service** и **sudo systemctl mask mdmonitor.service**.



► Сервисы *Systemd* управляются также графическим инструментом *Systemadm*.



0 и 100. Задайте более низкое значение, если хотите избежать перекачки процессов. При значении, равном 0, ядро будет избегать переноса процессов из физической памяти в раздел подкачки как можно дольше.

Команда `sudo sysctl vm.swappiness=10` временно изменит значение `swappiness` на 10. Это изменение будет потеряно при перезагрузке компьютера. Если вы заметите воздействие на свой дистрибутив, можете сохранить это значение, отредактировав файл `/etc/sysctl.conf` в текстовом редакторе. Найдите в файле параметр `vm.swappiness` и измените его значение. Если такого параметра нет, добавьте его в конце файла, вот так: `vm.swappiness=10`.

Создайте дополнительное ОЗУ

Если ваша проблема заключается в недостатке ОЗУ, данный прием создан для машин с достойным процессором, но небольшим объемом системной памяти — таких, как нетбуки и виртуальные машины. На таких компьютерах можно использовать модуль ядра `zRAM` для создания устройства в ОЗУ и сжатия его. Благодаря фактору сжатия, вы эффективно увеличите свое ОЗУ.

Учтите, что используя `zRAM`, вы нагружаете свой процессор, поскольку ему приходится все время сжимать и распаковывать. Тем не менее, это эффективнее и быстрее, чем переносить все на жесткий диск. Также помните: модуль `zRAM` считается стабильным начиная с ядра 3.14.

В Ubuntu вы можете установить модуль по `sudo apt-get install zram-config` и затем перезагрузить компьютер. Пакет устанавливает скрипт, чтобы запускать его в качестве сервиса, и не требует никакой настройки. Команда `cat /proc/swaps` выявит его наличие онлайн. Если `zRAM` работает хорошо, команда должна вывести один или более разделов `/dev/zram`. Если ваш процессор не способен справиться с дополнительной нагрузкой и удобство использования вашего компьютера под угрозой, можете отключить и удалить модуль `zRAM` с помощью `sudo apt-get purge zram-config`.

В Fedora для активации `zRAM` можно использовать скрипты из проекта `FedoraZram` GitHub (<http://bit.ly/FedoraZram>). Скачайте и распакуйте файлы проекта и перейдите в создавшуюся директорию. Затем установите инструменты Fedora, чтобы настроить свою среду компиляции, с помощью

```
sudo yum install @development-tools fedora-packager
```

После установки запустите `rpmdev-setuptree` для создания структуры директории, требуемой для сборки пакетов. Затем командой `make rpm` создайте RPM для сервиса `zRAM`. Когда пакеты будут созданы, можете установить их все с помощью `sudo rpm -Uhv ~/rpmbuild/RPMS/noarch/zram-*.noarch.rpm`. После установки включите сервис командой `sudo systemctl enable zram.service` и затем запустите его с помощью `sudo systemctl start zram.service`. Начиная с этого момента, можно использовать команду `zramstat`, чтобы проверить, нормально ли работает сервис.

Предзагрузка часто вызываемых программ

У большинства пользователей имеется некий набор приложений, которыми они часто пользуются. `Preload` — это настраиваемый демон чтения с упреждением, который работает в фоновом режиме, анализируя и отслеживая вашу работу с приложениями. На основе этого постоянного анализа `Preload` прогнозирует, какое приложение вы запустите следующим, находит все библиотечные зависимости и заранее загружает их в память. Это означает, что когда вы обратитесь к приложению, оно запустится практически мгновенно, поскольку уже сидит в памяти.

Например, если вы вскоре после загрузки дистрибутива всегда открываете `Firefox` и `LibreOffice Writer`, `Preload` автоматически загрузит их в память, когда ваш компьютер будет загружаться.

```
bodhi@bodhi-Lenovo-G505s: ~
bodhi@bodhi-Lenovo-G505s:~$ sudo e4defrag -c /dev/sda6 /dev/sda7
[sudo] password for bodhi:
<Fragmented files>
now/best      size/ext
1. /media/bodhi/korora/home/bodhi/.config/google-chrome/Default/Application Cache/Cache/Index      31/1      4 KB
2. /media/bodhi/korora/var/log/yum.log                       13/1      4 KB
3. /media/bodhi/korora/var/lib/gdm/.ICEauthority             10/1      4 KB
4. /media/bodhi/korora/var/log/wpa_supplicant.log-20140407  10/1      4 KB
5. /media/bodhi/korora/var/log/wpa_supplicant.log-20140521  10/1      4 KB

Total/best extents      289333/287308
Average size per extent  58 KB
Fragmentation score     0
[0-30 no problem; 31-55 a little bit fragmented; 56- needs defrag]
This device (/dev/sda6) does not need defragmentation.
Done.
```

Войдя в систему и запустив приложения, вы удивитесь, как быстро они грузятся. Но хотя это звучит заманчиво, все же `Preload` подходит не всем. Он отлично работает для тех, кто открывает несколько разных приложений; но тем, кто загружает приложения только время от времени, вряд ли будет полезен. В таком случае `Preload` будет без нужды занимать ОЗУ.

Некоторые дистрибутивы включают демон `Preload` по умолчанию, и почти во всех дистрибутивах он есть в официальных репозиториях. В Ubuntu он устанавливается по `sudo apt-get install preload`, а в Fedora — по `sudo yum install preload`.

`Preload` можно спокойно использовать с его настройками по умолчанию. А если вам нужно будет его настроить, подправьте конфигурационный файл демона (`/etc/preload.conf`) в текстовом редакторе.

Разгоните файловую систему

Подобно всему прочему внутри вашего дистрибутива Linux, файловая система тоже настраивается индивидуально. По умолчанию, разработчики дистрибутивов жертвуют производительностью ради безопасности, и хотя настройки по умолчанию прекрасно работают для большинства, отдельным индивидуалистам, возможно, захочется сменить их, с целью получения большей производительности.

«Если ОЗУ в вашей системе хватает, минимизируйте использование подкачки.»

Самая популярная файловая система, имеющаяся практически в каждом настольном дистрибутиве — файловая система `ext4`. Ее настройки по умолчанию должны подойти для большей части вашей работы, однако если ваши тесты производительности указывают на файловую систему как на слабое место, есть несколько способов ее настроить.

Первое, что надо сделать — отключить параметр `atime`. Когда этот параметр активен, при каждом обращении к файлу файловая система отмечает время данного события. Чтобы полностью от этого избавиться, откройте файл `/etc/fstab` и введите опцию `"noatime"` вместе с существующими опциями для всех разделов `ext4`.

Производительность файловой системы также можно повысить, включив индексирование директорий, которое ускорит чтение и запись из директорий. Однако используйте для включения индексирования директорий `live CD Linux`, и обеспечьте, чтобы раздел, для которого вы хотите его включить, не был монтирован. Предположим, ваш раздел `ext4` — `/dev/sda1`; тогда команда

» При большой интенсивности операций чтения/записи дефрагментируйте файловую систему инструментом `e4defrag`.

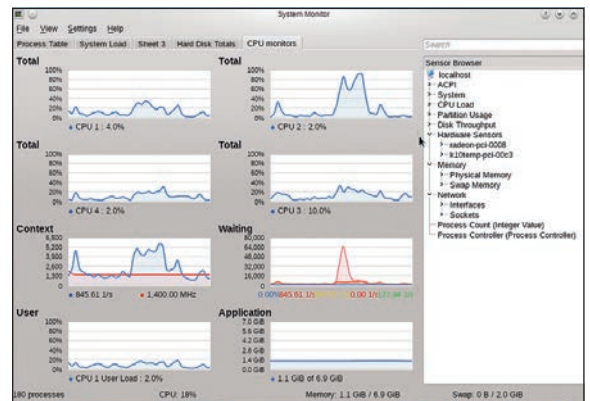
`sudo tune2fs -O dir_index /dev/sda1` включит эту функцию. Затем запустите `sudo e2fsck -D /dev/sda1` для индексирования существующих папок.

Индексирование директории является безопасной функцией для повышения производительности, а вот изменение режима журналирования файловой системы отнюдь не безопасно и может повредить вам файловую систему. Файловая система ext4 поддерживает три опции журналирования. Режим Journal — самый медленный, но исключительно безопасный. Затем идет режим Ordered, предлагающий разумный компромисс между скоростью и безопасностью; в большинстве дистрибутивов он используется по умолчанию. В этом режиме все данные вместе с метаданными перед записью в основную файловую систему сначала заносятся в журнал.

Третья доступная вам опция — режим Writeback, и именно ее вы должны использовать, если вас заботит производительность. В этом режиме данные записываются в основную файловую систему сразу же после фиксации их метаданных в журнале, что повышает производительность. Для изменения режима журналирования вашей файловой системы, загрузитесь в безопасную среду и затем запустите

```
sudo tune2fs -O has_journal -o journal_data_writeback /dev/sda1
```

чтобы переключить раздел `/dev/sda1` в режим записи.



► Встроенный в KDE System Monitor предлагает массу настраиваемых опций для мониторинга и решения проблем с производительностью.

```
sudo e4defrag -c /dev/sda1
```

чтобы проверить файловую систему на предмет фрагментации. Когда это будет выполнено, команда сообщит вам, необходима ли дефрагментация вашей файловой системе. Если необходима, то команда

```
sudo e4defrag /dev/sda1
```

дефрагментирует ваш раздел.

Заплатки на ядро

Если вам крайне нужна скорость, следует подлатать свое ядро. Для начинающих есть готовые заплатки [patch] ядра, оптимизированные для увеличения производительности.

Одну из самых популярных альтернатив стандартному ядру скомпилировал Кон Коливас [Con Kolivas] из множества разных заплаток. Его набор именуется -ck, и он создан с акцентом на увеличение производительности настольной системы. Когда вы его установите, вы заметите ускорение и при выполнении повседневных настольных задач, и при играх и воспроизведении мультимедиа.

Чтобы установить его, сначала через менеджер пакетов своего дистрибутива скачайте исходник ядра той версии, к которой относится патч, а также все инструменты для компиляции ядра. Затем скачайте патч с сайта Коливаса (<http://bit.ly/ConKolivas>) и употре-

бите команду **patch** в папке исходника ядра, чтобы применить изменения, например:

```
sudo patch -p1 < patch-3.*-ck1
```

Затем скомпилируйте ядро согласно инструкциям своего дистрибутива.

В Ubuntu проще будет использовать скрипт с <http://bit.ly/KernalckUbuntu>, написанный членами бразильского сообщества Ubuntu, который скачает вам простые ядра вместе с заплатками Коливаса и скомпилирует их в устанавливаемые бинарники. Скачайте скрипт в папку `/tmp` через посредство

```
cd /tmp; wget --no-check-certificate https://raw.githubusercontent.com/chilicuil/learn/master/sh/kernel-ck-ubuntu
```

и затем исполните с помощью

```
sh kernel-ck-ubuntu.
```

На выполнение всего этого уйдет некоторое время (может, чайку?). Когда все будет закончено, вы получите набор бинарников, устанавливаемых так:

```
sudo dpkg -i ./linux-*.deb.
```

«Заплатки ядра от Кона Коливаса: вы заметите ускорение повседневных задач.»

Дефрагментировать файловую систему Linux вам, как правило, незачем. Если кратко, то это потому, что популярные файловые системы Linux размещают файлы более разумно: не наваливая файлы один за другим, файловая система дает отдельным файлам достаточно места для роста. Но если вы читаете и пишете файлы в большом объеме и постоянно проверяете границы своего раздела, может понадобиться дефрагментация файловой системы. Если вы используете файловую систему ext4, то, возможно, ваш дистрибутив включает инструмент `e4defrag`. Предположим, ваши данные размещены в разделе `/dev/sda1`; тогда используйте



► Дистрибутив Elive на базе Enlightenment победил в нашем сравнении неприхотливых к ресурсам дистрибутивов (см. Сравнение, LXF186, стр. 22); он включает экспериментальные режимы для планшета и смартфона.

Достойные легковесы

Выберите из кучи облегченных дистрибутивов, менеджеров и приложений.

Популярные дистрибутивы с богатыми функциями приложениями — столпы современной системы Linux. Они улучшают вашу работу, а также охват аудитории и удобство использования дистрибутива. Однако алчность к ресурсам компьютера исключает их применение на более старых машинах с ограниченными ресурсами. Но благодаря богатству выбора и разнообразию приложений в мире открытого кода, альтернатив хватает, в том числе и специально предназначенных для более медленных машин. В дополнение к приемам, упомянутых на прошлых страницах, замените также некоторые приложения их легковесными аналогами, подстегнув старый компьютер.

» **Смените менеджер входа** Рассмотрите вариант легковесного менеджера входа или отображения, например, *XDM* вместо *LightDM*, *GDM* или *KDM*, поставляемого с вашим дистрибутивом. *XDM* менее красив, зато его требования к оборудованию минимальны.

Поскольку он имеется в официальных репозиториях большинства дистрибутивов, его можно установить через менеджер пакетов своего дистрибутива. После установки проверьте, что файл `/etc/X11/defaultdisplay-manager` показывает полный путь к менеджеру отображения *XDM* (обычно `/usr/bin/xdm`). При последующих загрузках вас встретит аскетичный *XDM*.

» **Используйте более легковесный менеджер окон** Нынешнее поколение популярных композитных менеджеров окон, таких, как *Mutter*, *Compiz* и *Kwin*, очень требовательное. На старых машинах вы можете отключить их и использовать вместо них одну из легковесных альтернатив, например, *OpenBox*, *Fluxbox*, *FLWM* и т. д.

Очень популярная опция *OpenBox* есть в официальных репозиториях большинства дистрибутивов. Помимо пакета *openbox*, установите также *obmenu* (редактор меню), *obconf* (инструмент настройки), *tint2* (легковесную панель задач) и *gmrn* (легковесную программу запуска приложений). После установки создайте его директорию настройки в своей домашней директории с помощью `mkdir ~/.config/openbox` и скопируйте сюда файл настройки по умолчанию из его исходного местоположения в `/etc/xdg/openbox/rc.xml`. Затем создайте файл автозапуска (`~/.config/openbox/autostart`), который *OpenBox* будет считывать при каждом вашем входе. Добавьте команду `tint2 &` в новой строке внутри файла, чтобы автоматически запускать панель задач.

» **Задействуйте рабочий стол в весе пера** Если вы используете более старое оборудование, которое плохо тянет рабочий стол Ubuntu по умолчанию, Unity, вам, возможно, нужен рабочий стол полегче. И опять же, вам есть из чего выбирать, включая *Mate*, *Xfce*, *LXDE* и *Enlightenment*. *LXDE*, в частности, содержит основные функции для урезанной, но вполне функциональной среды рабочего стола. Там нет избытка блестящих графических эффектов или ненужных функций, путающихся у вас под ногами.

В Fedora *LXDE* и его утилиты устанавливаются по `sudo yum install @lxde-desktop`, а в Ubuntu — по `sudo apt-get install lxde`. В качестве альтернативы можете установить индивидуально настроенный рабочий стол *LXDE* из проекта Lubuntu, по `sudo apt-get install lubuntu-desktop`.

» **Применяйте специдистрибутивы** Если нужно быстрое решение проблемы повышения скорости старых компьютеров, то вместо замены отдельных приложений вы можете подойти к делу более основательно и установить легковесный дистрибутив. Есть несколько очень симпатичных — таких, как Lubuntu, Bodhi, CrunchBang и Puppy Linux.

Эти дистрибутивы из всех сил стараются не переутомить ваше оборудование. Например, Puppy Linux имеет множество легковесных индивидуально настраиваемых приложений всех видов, а в дистрибутиве Lubuntu имеется модуль ядра *zRam* для улучшения производительности на машинах с малым объемом ОЗУ.

Легковесы на рассмотрение

Bodhi — один из лучших дистрибутивов, разработанных для компьютеров с малыми ресурсами. Он использует легковесный и приглядный рабочий стол *Enlightenment* и предлагает ряд профилей, в том числе Bare, Laptop и Fancy; каждый профиль оптимизирован под разные типы настройки оборудования. По умолчанию дистрибутив предлагает минимальный набор приложений, но вы можете добавить приложения с помощью сетевого инструмента установки программ *AppCenter*.

Затем идет Lubuntu; это официально поддерживаемая версия Ubuntu на базе легковесного рабочего стола *LXDE*. В отличие от Bodhi, Lubuntu содержит массу приложений. Пакетами в дистрибутиве управляет собственная легковесная версия *Ubuntu Software Center*. Можете также обратить внимание на другие дистрибутивы на базе *LXDE*, например, Linux Mint *LXDE* и WattOS.

«Замените приложения легковесными аналогами, подстегнув старый компьютер.»

Если вы не против попробовать нечто экзотическое, есть Puppy Linux — постоянный фаворит среди легковесных дистрибутивов. С 2003 года он превращает непригодные к использованию компьютеры в сущие Феррари. У Puppy Linux один из самых легких менеджеров окон (*JWM*), и хотя этот дистрибутив не самый привлекательный внешне, он все же полон индивидуально настраиваемых приложений и предъявляет очень низкие системные требования.

И, наконец, есть дистрибутив CrunchBang, с менеджером окон *OpenBox*. Разработчики не позиционируют его как дистрибутив для медленных компьютеров, что объясняет включение в его состав довольно мощных приложений вроде *LibreOffice*. Однако несмотря на состав его приложений, этот дистрибутив не требует столько ресурсов, как популярные дистрибутивы рабочего стола, что делает CrunchBang идеальным для полузабытых машин. **LXF**



Легковесные приложения-аналоги

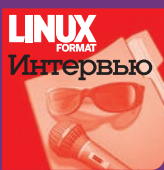
Вот список приложений, облегченных как по функциям, так и по объему — они могут положительно отразиться на производительности старых машин. [Полноценный список вы найдете в нашем материале, посвященном приложениям с низкими ресурсами, **LXF186**, стр. 22 и 28].

- » Текстовый процессор AbiWord
- » Электронные таблицы Gnumeric
- » Просмотр изображений Feh
- » Электронная почта Claws Mail
- » Браузер Midori
- » Видеоплеер VLC Media Player
- » Просмотрщик PDF Xpdf
- » Обмен быстрыми сообщениями Pidgin
- » Клиент BitTorrent Transmission
- » Менеджер файлов Worker



Open Education

Linux Format открывает, как Open Source и открытые платформы формируют будущее образования в Великобритании и как это воплощается на практике.



Образование в Великобритании, как и во всём мире, меняется. Появление объединённых открытых платформ, а также Massive Open Online Courses (MOOC)

и специфически британское стремление вернуть программирование в школы стимулировали новый всплеск интереса к технологиям по всему миру. **Linux Format** посетил форум экспертов в области образования ранее в 2014-м и разузнал, как Open Source может, и может ли, изменить будущее образования.

Марк Чемберс [Mark Chambers]

Etag и UKForCE (Форум по компьютерной грамотности)

LXF: Задача Etag — консультировать правительство, как сделать программирование весёлым и увлекательным?

МЧ: Технологии можно использовать и за рамками учебного плана по информатике. Etag — это инициативная группа по разработке образовательных технологий, созданная по решению правительства, чтобы, так сказать, задавать тон. В их задачи не входит создание стандартов,

ограничений или правил — только продвижение среди политиков идеи широкого использования технологий.

LXF: В чём Etag видит роль Open Source в этом процессе?

МЧ: Мы в восторге от технологических решений, целесообразных в плане цены, сложности и исполнения. И я считаю, что без Raspberry Pi и Open Source здесь не обойтись. Недавно нам удалось посодействовать TLM в разработке квалификации 1-го, 2-го и 3-го уровней в Open System Computing. Что дополнит более традиционный GCSE



ОБ ОБУЧЕНИИ СОТРУДНИЧЕСТВУ «Бессмысленно штамповать отличников по математике, неспособных сотрудничать.»

[обязательные экзамены по окончании средней школы в Великобритании, — прим. пер.] по информатике. Проблема, однако, в том, что школы — разборчивые покупатели. И сперва надо убедить учителей, что этой системе можно доверять.

LXF: Что же тогда сдерживает внедрение Open Source и открытых стандартов?

МС: Во-первых, проблема в отсутствии надёжного канала связи со школами. Всё чаще нет ответственных за это в органах местного самоуправления; всё больше и больше школ вынуждены сами обращаться к нам за помощью. Мы можем указать им, где её можно получить, но мы — организация небольшая, и напрямую всем помочь не можем. Вероятно, в сообществе есть ещё много энтузиастов, находящихся в таком же положении.

LXF: Но нельзя же просто доставить в начальную школу несколько Raspberry Pi и учебник по Linux, сказав «удачи вам».

МЧ: Многие материалы бесплатны, но повышение квалификации, которое должно вестись

параллельно, обходится очень дорого — это не только подготовка учителей, но и оборудование учебных кабинетов, консультирование и наставничество, необходимые для обеспечения эффективности.

LXF: Какие плоды это может принести через 10 лет?

МЧ: Я не из тех, кто считает, что школы нам будут не нужны. Я глубоко убеждён в необходимости личного сотрудничества. Но мне кажется, обучение [могло бы] вестись под большим контролем самой молодежи, и не только локально, но и в глобально, с использованием уже известных нам инструментов. Вряд ли существуют такие дистанционные курсы [МООС], в основе которых не лежали бы традиционные парадигмы. Но есть и новые подходы к обучению, уже применяющиеся в школах, которые также можно использовать в виртуальных сообществах или хотя бы взять на заметку. В любом случае, как физически, так и виртуально, они должны применяться с умом.

LXF: Насколько сегодня Etag готов к обучению программированию не только 3%, но и всех 100% детей?

МС: Мы глубоко убеждены, что существуют аспекты обучения технологии, интересные всем молодым лю-

дям. И я бы не сводил это только к использованию; скорее, дело в объединяющем начале, в том, для чего это делается. Это может быть социальное предпринимательство, средство помощи людям с ограниченными возможностями. Какие средства? Аппаратные и программные, помогающие это осуществить. И по сути, заниматься этим нужно не индивидуально, а сообща. Так что я согласен с тем, что существует потребность расширить сферу компьютерного знания за пределы узко-специального, но мне бы хотелось, чтобы упор делался именно на коллективную деятельность.

LXF: Многих родителей удручает, что учебный план ограничивается лишь изучением электронных таблиц.

МЧ: Меня поражает эта раз за разом повторяющаяся фраза «мой ребёнок ничего, кроме текстового редактора или базы данных, не проходил». Она заостряет внимание на том, что этому неправильно учили. Но в то же время есть и прекрасные примеры учения и преподавания, которым следует радоваться. Похоже, в некоторых школах для

неблагополучных детей, которых ни за что не усадишь за компьютер, выработали очень креативные практические методики обучения, вроде телестудий или создания игр. Такое всегда было, и мы должны признавать это, а не думать, что начинаем всё с нуля. О-о, вот я и вспомнил Scratch! Надеюсь, что удастся эту тему обойти.

LXF: Ни за что! Как можно внести туда свою лепту?

МЧ: Возможностей становится всё больше. Многие члены Naase принимают участие во внешкольных мероприятиях. Так, например, один из наших членов в Лондоне при поддержке Hayes Recruitment организует Coder Dojo, и участие примет сотня молодых людей.

LXF: А как удаётся привлечь молодёжь обоих полов?

МЧ: Честно говоря, понятия не имею. Важно, чтобы у всех были равные шансы, вне зависимости от пола или чего-либо ещё. Это острая борьба. Вероятно, мы недостаточно осведомлены о методах, пригодных для разных сообществ. Девочкам важнее практическая сторона программирования, нежели собственно разработка кода. Но это лишь свидетельства учителей, которые мне доводилось слышать. Думаю, нам нужен оперативный отклик, основанный на фактических данных. Пока у нас его нет.

LXF: Мы — поколение тех, кто смог, имея 8-битные домашние компьютеры, завоевать мир игрового программирования, и того человека, который изобрел Интернет.

МЧ: Учителя сыграли в этом большую роль — они сами из программистов на BBC B / Archimede и ушли в разнообразные предприниматели. Если вы спросите, что же произошло, то это из той же серии, что про курицу и яйцо: может, это мы их так прижали, что у них не стало времени на подобную творческую деятельность? Кто-то сбил их с толку, и зачастую мы виним в этом национальную политику. Это всё пресловутое приведение к общему знаменателю.

LXF: И куда это приведёт страну через 10 лет?

МЧ: Есть мнение, что больше всего рабочих мест будут создавать не крупные, динамично развивающиеся организации. Думаю, это как раз связано с тем, о чём я говорил ранее — что учебный план должен быть нацелен на создание небольших сообществ, где молодёжь сможет объединяться в команды и заниматься дизайном/разработкой/контрактами для решения реально существующих проблем конкретной целевой аудитории. Мы примеряемся к тому стилю деятельности, который, как считается, будет востребован в будущем. Все сходятся на том, что бессмысленно делать ставку на количество и штамповать отличников по математике, неспособных работать вместе или сотрудничать ради создания чего-либо. Гораздо лучше применять более сбалансированный подход к обучению, который будет способствовать предпринятиям малого бизнеса.



Клайв Бил [Clive Beal]
Raspberry Pi Foundation

LXF: Pi создавался как неформальный инструмент в ответ на неспособность образования охватить одновременно программирование и Open Source. А теперь является официальным средством обучения и тому, и другому. Вот это успех!

КБ: Так и есть, мы это понимаем, и наша образовательная команда плотно работает над написанием ресурсов. Мы осознаём переход от неофициального к официальному; это не входило в наши исходные планы, но мы этому рады. Я сам бывший учитель, и вижу, что это очень полезный инструмент. Вернее, часть инструментария. Мы никогда не считали его единственным решением проблемы: Pi входит в арсенал образовательных средств, с которыми можно побаловаться и поэкспериментировать. Мы готовимся открыть CPD (Continuing Professional Development) [программа повышения квалификации, — прим.пер.] для учителей, где мы будем создавать ресурсы и куда учителя смогут прийти, чтобы использовать их по всей форме.

LXF: Вы ориентируетесь на начальное и среднее образование?

КБ: Возможно, и на начальное — где-то между вторым и третьим Ключевым Этапом [Национальной программы обучения Великобритании, — прим. пер.] Мы также ведём работу над GCSE. У нас есть онлайн-курс совместно с Cambridge University Press, OCR, например. Мы отлично знаем, что учителя начальной школы — не специалисты. Но есть вероятность, что уже в сентябре 2014 им придётся вести этот совершенно новый для них предмет. И после посещения моих курсов они думают: «это

круто, и это мне по силам». Но сначала все приходят с мыслью, что программирование — это просто тёмный лес.

LXF: Есть опасения, что для проведения урока программирования раз в неделю учителям требуется море подготовки. Ваши курсы помогают?

КБ: Что касается курсов, что вёл я сам, вскоре появится CPD — она сейчас в разработке. Для меня главное то, что на занятиях мы полдня вообще не садимся за компьютеры — а это однодневный курс. Мы занимаемся CS Unplugged — учим

О ФИЛОСОФИИ OPEN SOURCE

«Превратить проповеди над заблокированными коробками в исследование.»

информатику без компьютеров. Акцент делается на принципах, лежащих в основе всего. Организация сети, абстракция, алгоритмы.

LXF: В 70-х, когда компьютеры школам были не по карману, вы бы обучали блок-схемам и основам логики.

КБ: Я мог бы отвести школьников на игровую площадку и там поизучать блок-схемы, создать маленьких роботов, порешать задачки. Так что способности есть. Люди любят разгадывать загадки и создавать всякие штуковины. Так что учителя начальной школы справляются с «алгоритмами» и прочими диковинными словами на раз. Они понимают, что это здорово и весело, и дети это любят. А насчёт того, что детям не объяснить,

что такое алгоритмы — вы же учите их читать, а не проповедуете необходимость грамотности, только с программированием... Я понимаю эти опасения, нам это очень понятно. Учителей начальной школы мы хотим и будем поддерживать.

LXF: В каком объёме вы хотите внедрить принципы свободного ПО?

КБ: Учебный план по ИКТ, в том виде, в каком он был, позволял делать кучу невероятных проектов, но все они были в коробках под замком, как в физическом, так и программном смысле. Нельзя было прийти в техподдержку и сказать: «Включите, пожалуйста, сетевое ПО». Не так всё просто. В корпоративном отношении, многое ограничивалось лицензиями Microsoft или другими — естественно, закрытыми — лицензиями. И, опять же, всё зависело от учителя. В том смысле, что стоило только рассказать о существовании таких великолепных инструментов, как *Audacity*, показать его детям, и вот они уже пишут музыку со своих же сэмплов. О Scratch — не знаю ни одного учителя 7-го класса, который не использует Scratch, чтобы в каком-то смысле и развлечь детей в ходе обучения программированию, и провести контроль. Так что вместо блок-схем теперь можно делать то же самое в Scratch. Чем больше людей о нем [Open Source] знают, тем лучше, но нам надо выйти из этих рамок, а в начальных школах проблема зачастую в том, что у них либо один координатор по ИКТ, который приходит раз в неделю, либо его вообще нет, и тогда они идут к ближайшему учителю физики и говорят: «Ты! Вот ты этим и займёшься». Нет такого, что можно найти потрясающий инструментарий, поставить его на компьютер и дать им поиграть.

LXF: Есть ли у вас связи со старшей школой, чтобы помочь им поддерживать начальную?

КБ: Они должны быть, и нужен механизм взаимодействия во всех школах, но он не всегда есть. Когда мы только набирали наш первый курс, это было всё равно что продавать двойные стеклопакеты, но с тех пор всё сильно изменилось, ведь

теперь они понимают, что без этого не обойтись. И мы просто обязаны показать им, насколько это творческое и увлекательное занятие, что это вовсе не то скучное программирование, о котором разглагольствуют правительство, политики и СМИ. Именно это их и пугает, они думают: «К сентябрю я должен стать гениальным программистом». А речь не о том.

LXF: Начальная школа всегда лучше умела превратить учение в развлечение, чем старшая.

С появлением клубов программирования это стало проще, но ведь они есть не везде.

КБ: Как только учителя поймут, что это им по силам, и в особенности, что компьютер дети осваивают практически сами, к ним придёт уверенность,

что это прекрасный предмет, к тому же междисциплинарный — стоит попробовать.

LXF: Чем наши читатели могут помочь в работе над учебным планом и поддержке школ?

КБ: Думаю, участие в Code Club и Code Club Pro, если вы ИТ-специалист — лучшее, что вы можете сделать как родитель. Но можно помочь, и не имея специальных знаний. А именно, если вы подумываете о Pi — то они довольно дешевы.

LXF: Родители — не технари могут поучаствовать в сборе средств. Не успеете оглянуться, как в вашей школе появятся штук пять симпатичных компьютеров на Linux!

КБ: Ещё можно обратиться к местным предпринимателям. Кэрри Энн [Филбин] [Carrie Ann Philbin] нужна была техника для школы, она написала об этом в Твиттере, и уже через полчаса нашёлся желающий отдать ей мониторы. Нам нравится и скидываться всем миром, и мастерить. Можно сделать в уголке лабораторию, имея всего несколько дешевых роботов, несколько Raspberry Pi и несколько Arduino. Лично я хотел бы превратить проповеди над заблокированными коробками в исследование, когда каждый может вникнуть и получить удовольствие. Изменить эту атмосферу в классе будет почти так же сложно, как и учебный план.

LXF: Вы посещали Raspberry Jam?

КБ: Это просто откровение, начиная с самих волонтеров, которые говорят: «Я люблю это, это важно, я хочу поведать об этом миру». У тебя открываются глаза на то, что люди готовы это делать — сообществу, в котором уже свыше 90 000 человек, такое по силам. Сообщество огромное, такое дружелюбное и на самом деле полезное. И вот что значит — в нужное время, в нужном месте. Во многом параллельное новому учебному плану, это мероприятие увлекло множество людей своим творческим духом. Мы за это очень благодарны.

LXF: Стало быть, информатика больше не скучный и занудный предмет!

КБ: В том и суть! И кроме того, междисциплинарный — вплоть до искусства и спорта; но самые тесные связи у него с наукой и музыкой. Они так и ждут, чтобы их задействовали. И это самое увлекательное. Можно привязать сюда хоть гроб сестры Аддамс, хоть высотный аэростат — я знаю школы, где это уже делают. Раньше на это бы ушло £700, а теперь можно купить Pi, добавить к нему дешёвый GPS и отправить его наверх всего за пару сотен фунтов. И сделать что-то невероятное, например, снимки из космоса, не прилагая никаких особых усилий. Кроме того, есть Sonic Pi, разные штуковины для музыкантов и художников. У нас есть Рейчел [Rachel], наш штатный художник, и это прекрасный пример для девочек. Возвращение информатики в учебный план открывает огромные перспективы; по-моему, это очень здорово, и я горжусь, что имею к этому отношение.

Раджай Наик [Rajay Naik]

Open University [Открытый университет]

LXF: Какова позиция OU относительно влияния высшего образования на школьные учебные планы?

РН: Почему, собственно, какой-нибудь старшеклассник в свои 18 лет не может получить доступ к тому же общемировому уровню образования, что и обладающий статусом студента? Благодаря Интернет-образованию, какое осуществляет Khan Academy и многие другие, теперь становится возможным составить индивидуальный учебный план, сообразно амбициям и пожеланиям любого учащегося.

LXF: И как это в итоге будет происходить?

РН: Примерно так, как Blockbuster пришлось объявить о своём банкротстве с появлением Netflix, Lovefilm и в связи с цифровой революцией. По сути, настает момент, когда технология позволяет аудитории обойти руководство системой образования. Речь не о том, что бывшие кампусы распродадут под торговые центры, но мы станем свидетелями радикальных перемен в осуществлении обучения, и это хорошо.

LXF: Вы сказали, что в британских университетах МООС приживаются медленнее, чем в американских. Значит ли это, что в британской сфере образования сопротивляются инновациям?

РН: На самом деле, зачастую даже хорошо не быть в первых рядах. Например, мы сперва создали Future Learner для мобильных и планшетов, а не для настольных ПК, как в Америке. На самом деле, очень ободряет то, как далеко нам удалось продвинуться за каких-то 15 месяцев. Мы начали с идеи подключить 29 мировых университетов

к первой в Европе системе дистанционного образования, и дошли до курсов, доступных сотням тысяч студентов из более чем 190 стран, причём всего за год.

LXF: Какую роль здесь играет Open Source?

РН: Думаю, институты понимают, что правила игры изменились, и теперь очередь за ними. Но в основе всего должно быть качество. Так, на некоторых платформах вам предоставляют только лекцию, записанную на камеру, стоящую в конце аудитории — это не создаёт эффекта присутствия и потому непригодно. Наша цель — качество, а не количество.

LXF: Существует представление о том, как будет выглядеть образование к 2025 году. Куда приведёт нас использование открытых платформ и открытого кода?

РН: В отношении образования и технологии, Британия по-прежнему в списке лидеров. Мы занимаем второе место в сфере высшего образования на планете — это приносит нам £11 миллиардов экспорта в год. Так что для нас главное — сохранить эти позиции в последующие 30 лет.

LXF: Как, по-вашему, Raspberry Pi вписывается в концепцию Open University?

РН: Именно такое нам и нужно! Самое главное — это использовать имеющиеся технологии для создания лучших продуктов и сервисов. И здесь простота — ключевой момент. Raspberry Pi — простой инструмент, именно к этому мы и стремимся. Мы хотим создать простую среду, которая изменит жизни людей. Смотрят ли BBC, используют iTunes, YouTube, OpenLearn или получают степень в OU — если это можно сделать просто, мы на правильном пути. **LXF**





Состояние искусства 4K

Ультра-высокое разрешение — «очередная крутая штука», но хорошо ли 4K поддерживается в Linux? Мэтт Хэнсон пытается это выяснить.

Кривая эволюции разрешения мониторов отнюдь не идет все время вверх, как в большинстве отраслей технологии.

В старые добрые времена увесистых бежевых мониторов — которые справлялись с задачей обогрева наших домов ничуть не хуже, чем с основной задачей отображения рабочего стола — существовала устойчивая тенденция роста максимального разрешения по мере появления

новых моделей. В 1990-е одним из самых популярных было разрешение SVGA 800×600 (достойное улучшение по сравнению с разре-

«Помимо фильмов и изображений с эффектом присутствия, 4K дает обширное рабочее пространство.»

шением VGA 640×480), а некоторые CRT дошли аж до разрешения 2048×1536. С появлением плоскоэкранных мониторов быстро стало

стандартом разрешение 1920×1080, известное также как «full HD», после чего развитие разрешения несколько застряло. И хотя мы наблюдали

недавний всплеск активности в данной области, отчасти вызванный наличием доступных по цене мониторов с разрешением 2640×1440 (и дисплеев Apple «Retina» с разрешением до 2880×1800), именно разрешение 4K (3840×2160) по-настоящему взбудоражило людей.

Масштабные фильмы и изображения с эффектом присутствия — не единственное преимущество 4K. С практической точки зрения, 4K дает обширное рабочее пространство, позволяя одновременно отображать многочисленные приложения и окна без дополнительного монитора.

Единственным препятствием была стоимость. Еще недавно, желая приобрести монитор 4K, вы упирались в цену не менее £1000. К счастью, рынок уже наводнен мониторами 4K от таких солидных фирм, как Philips, AOC, Asus и Samsung, по более разумным ценам: примерно £500–600. Два отличных 28-дюймовых монитора — Asus PB287Q за £580 и AOC U2868PQU примерно за £499. Возможно, они отчасти растягивают понятие «разумной цены», но это определенно шаг в верном направлении, и дисплеи 4K будут все дешевле.

Другой крупный расходный пункт 4K — оборудование для работы с ними; впрочем, требуемая мощность машины будет определяться тем, что вы хотите делать со своим 4K. Просто для расширения пространства рабочего стола и визуальной точности фотографий запросы к оборудованию на диво скромны. При современных материнской плате и процессоре вам, возможно, не понадобится даже отдельная видеокарта. С 4K ладит Zotac mini Zbox E1730 Plus [см. Обзоры, **LXF187**, стр. 22], и нам удалось подключить к дисплею 4K даже Chromebox.

4K по умолчанию

Для просмотра и редактирования на 4K мультимедиа и игр нужна мощная видеокарта — или несколько карт — с портами DisplayPort 1.2, способными работать с 4K и частотой обновления 60 ГГц. HDMI для разрешений 4K применять допускается, но на данный момент HDMI 1.4 поддерживает 4K только при частоте обновления 30 Гц, а это не идеально для игр и приставок DVI на 1440p. HDMI 2.0 уже будет справляться с 4K.

Если у вас есть нужное оборудование и относительно новый дистрибутив, подключение монитора 4K не будет особо отличаться от любого другого монитора. Подключив его к своему

ПК, вы должны суметь загрузить свой дистрибутив Linux как обычно, однако рендеринг вашего рабочего стола теперь будет при разрешении 4K, 3840×2160. Тогда вы увидите, что значки, курсоры мыши и прочее станут довольно мелкими.

Вы можете узнать текущее разрешение, введя в терминале **xrandr**. Загрузится инструмент **xrandr**, используемый для настройки разрешения вашей системы. Запустив **xrandr**, вы увидите информацию о дисплее, например:

```
Screen 0: minimum 320 x 200, current
3840 x 2160, maximum 16384 x 16384
DFP1 connected 3840x2160+0+0 (normal left
inverted right x axis y axis) 621mm x 341mm
```

При слове «current» указано разрешение 3840×2160, то есть 4K отображается правильно. Ниже вы должны увидеть список доступных разрешений с частотой обновления. Там, где указано «DFP1 connected», это ссылка на порт, к которому подсоединен ваш монитор. Если ваш монитор на данный момент не дает 4K, с помощью **xrandr** можно изменить разрешение, введя

```
xrandr --output DFP1 --mode 3840x2160
```

Замените **DFP1** соответствующим именем вывода для вашей видеокарты. Вы также можете изменить частоту обновления монитора, прислав параметр **--rate** и требуемую частоту обновления. Например, вот так подключается 4K с частотой обновления 60 Гц:

```
xrandr --output DFP1 --mode 3840x2160 --rate 60
```

Потеха с драйверами

Если у вас есть все необходимое оборудование, но 4K все равно не воспроизводится, первым делом нужно убедиться, что не устарели графические драйвера. Чтобы легко и быстро выяснить производителя и модель видеокарты и версию драйвера, установленного на вашем компьютере, воспользуйтесь инструментом **glxinfo**. Возможно, сначала вам придется установить его из терминала, посредством

```
sudo apt-get install mesa-utils
```

Установив его, введите

```
glxinfo | grep OpenGL
```

пользователя и паролем, затем удалите все имеющиеся драйверы:

```
sudo apt-get purge nvidia*
```

Для карты AMD/ATI введите «ati» вместо «nvidia». Если Ctrl+Alt+F1 просто выведет вам пустой экран, нажмите Ctrl+Alt+F7, чтобы вернуться в графический интерфейс, затем введите

```
sudo sed -i -e 's/#GRUB_TERMINAL/GRUB_TERMINAL/g' /etc/default/grub
```

```
sudo update-grub
```

Перезагрузитесь, затем остановите текущую сессию X с помощью:

```
sudo service lightdm stop
```

Если это не сработает, попробуйте



► Sapphire AMD R9 290X предлагает отличное соотношение цена/мощность.

и получите информацию о вашей видеокарте и драйвере. Среди прочей выведенной информации будут две строки примерно такого вида:

```
OpenGL renderer string: AMD Radeon R9
290 Series OpenGL version string: 4.3.12618
Compatibility Profile Context 13.251
```

После слов «OpenGL renderer string:» будет указана ваша видеокарта, а после слов «OpenGL version string:» вы увидите версию своего драйвера, и вам станет ясно, установлены ли самые свежие драйверы. Пусть это не всем понравится, но мы на данном этапе все же рекомендуем для наилучшей поддержки 4K позаботиться о наличии проприетарных драйверов с закрытым кодом от производителя, а не искать их открытые альтернативы. Для видеокарт Nvidia загляните на <http://bit.ly/lxfnvidia>, для AMD — на <http://bit.ly/lxfAMD>, а если у вас чип процессора Intel со встроенной картой, загляните на <http://bit.ly/lxfintelgraphics>.

Конечно, альтернативы драйверов с открытым кодом, созданные и поддерживаемые сообществом энтузиастов, на свете есть; и если вы ратуете за открытый код, приглядывайте за такими проектами, поскольку, надо надеяться, поддержка 4K скоро улучшится. Для драйверов Nvidia с открытым кодом загляните на сайт »

Устанавливаем графические драйверы

1 Скачайте свежие драйверы Для начала, перейдите на официальный сайт производителя своей видеокарты (здесь мы установим драйвер Nvidia). Найдите самую свежую версию драйвера для вашей видеокарты и скачайте ее. Если драйверы будут в сжатом файле, распакуйте их в папку **Downloads**. Затем откройте терминал и сделайте файл драйверов **.run** исполняемым, командой:

```
chmod 777 ~/Downloads/NVIDIA-Linux-*-340.24.run
```

Замените «NVIDIA-Linux-*-340.24.run» на имена установленных вами драйверов.

2 Долой старое, даешь новое Теперь выйдите из графического пользовательского интерфейса по Ctrl+Alt+F1 (или F2). Войдите под своим именем

```
sudo service gdm stop
```

или, возможно, **kdm** вместо **gdm**.

3 Ставим новый драйвер Теперь введите

```
sudo sh ~/Downloads/NVIDIA-Linux-x86_64-340.24.run
```

и следуйте инструкциям по установке. Снова убедитесь, что вы ввели правильное имя файла для файла **.run**, который вы собираетесь установить. Затем введите

```
sudo reboot
```

для перезагрузки. Ваш новый драйвер должен быть установлен, и если вы проверите свои настройки системы, он должен отображать правильную модель и изготовителя видеокарты.

<http://nouveau.freedesktop.org>, а для AMD — на <http://xorg.freedesktop.org/wiki/RadeonFeature>.

Установив самые свежие драйверы Nvidia на нашу систему 4K, мы обеспечили поддержку двух видеокарт Titan Black с помощью `sudo nvidia-xconfig -sli=auto`

Ubuntu, Gnome и 4K

Конечно, заставить свой дистрибутив Linux работать с 4K будет хорошо и мило, но это вовсе не означает особого удобства работы. С нереально высоким разрешением многие шрифты, значки и элементы интерфейса станут очень мелкими. Это может смотреться вполне симпатично, но затруднит вашу работу с ОС, поскольку значки муравьиного размера требуют филигранной точности щелчков. Осмотр крошечных элементов интерфейса тоже заставляет напрягать глаза, так что мы запустили на 4K ряд дистрибутивов, чтобы посмотреть, как их рабочие столы по умолчанию справляются с высоким разрешением.

Первым был Ubuntu 14.04. Хотя его рабочий стол Unity не всем по нраву, с 4K он работает отлично. Собственно, мы бы даже сказали, что интерфейс пользователя выигрывает от высокого разрешения 4K. Интерфейс с поисковыми линзами, который частенько замусоривал экран, теперь привязан к верхнему левому углу экрана, предоставляя вам куда лучший обзор ваших программ и файлов и оставаясь весьма ненавязчивым. Чуть больше по размеру значки в менеджере файлов — явный результат стремления Canonical сделать Ubuntu удобным для использования на устройствах с сенсорным экраном — достаточно велики, чтобы использовать их при таком разрешении. Вероятно, лучше всех боковая панель Launcher, где размещаются значки ваших приложений. В 4K для этого прибавилось места, что сильно повышает удобство использования Ubuntu. Если эти значки слишком мелкие, вы легко сможете изменить их размер, зайдя в System Settings, выбрав Appearance и переместив бегунок размера значков в Launcher, чтобы добиться нужного размера.

Если вы категорически против Unity, имеются и другие опции. Прежде чем выйти из Ubuntu, мы установили оболочку и среду рабочего стола Gnome, чтобы посмотреть, как он справится с 4K по сравнению с Unity, и в некоторых областях



► Мониторы с 4K: обширное рабочее пространство с текстом, мелким до нечитаемости.

разница была огромной. Например, открыв меню Activities и выбрать Show Applications, вы увидите список своих приложений, и хотя значки крупные и красивые, но шрифты исчезающе мелкие и сильно затрудняют чтение. Вне этого меню все несколько проще, однако некоторые значки и меню страдают по сравнению с более крупными размерами по умолчанию в Unity. Релиз Gnome 3.12 показал, что он принимает 4K всерьез.

Xubuntu, Xfce и Mint

Xubuntu — это производное Ubuntu с рабочим столом Xfce. Лежащий в основе дистрибутив в основном тот же; различия велики только из-за разных рабочих столов. В отличие от Unity, вполне понятно, что Xfce был разработан для более привычной настольной работы, и хотя тут нет ничего плохого, это означает, что при разрешении 3840×2160 значки, шрифты и меню выглядят очень мелкими. Работать можно, но куда менее удобно, чем в Unity.

Затем мы попробовали Linux Mint. Рабочий стол этого дистрибутива, Cinnamon, приобрел большое количество поклонников — в особенности среди новичков в Linux — благодаря его сходству со структурой Windows (до версии 8), с Панелью задач внизу и меню Start (которое Mint называет просто Menu), предлагающим простой доступ к файлам и папкам. Так же, как и в Xfce,

в Cinnamon шрифты и значки кажутся мелкими и затрудняют использование Menu. К счастью, это легко исправить, щелкнув правой кнопкой по панели задач внизу (в Linux Mint именуемой Panel), выбрав Panel Settings и затем увеличив размер пикселей до высоты Bottom panel. Mint соответственно изменит размер значков и шрифтов, значительно облегчив использование.

Elementary и Fedora

Возможно, Elementary и не захватил столько же компьютеров, сколько Ubuntu или Mint, но быстро набирает популярность — во многом благодаря поставке с собственной средой рабочего стола и оболочкой под названием Pantheon, разработанной ради скорости и эффективности, да еще и эффектного вида. С 4K, Elementary OS прекрасно смотрится со своей стильно-минималистской средой Pantheon, которая отлично ладит с увеличившимся благодаря высокому разрешению рабочим пространством. Программа запуска приложений



► Bitfenix Colossus Window соответствует своему названию, однако избыток пространства внутри означает отличную вентиляцию, спасающую нашу установку 4K от перегрева.

Наши машины с 4K

Для наших тестов мы использовали две машины. Первой был довольно стандартный настольный ПК со средними спецификациями, включая малость устаревший на сегодняшний день Intel Core i7-2700K CPU. Мы тестировали на нем 4K, используя как встроенную видеокарту, так и высококлассную видеокарту Sapphire AMD R9 290X (www.sapphire-tech.com), разработанную для игр 4K. Нашей второй машиной был сносшибательный ПК, созданный для 4K и оснащенный мощнейшими из имеющихся компонентов, благодаря любезности Zotac (www.zotac.com) и Palit

(www.palit.biz) — включая не одну, а две видеокарты Nvidia Titan Black, размещенных в удобных больших шасси Bitfenix Colossus Window (www.bitfenix.com). Энергоснабжение таких мощных компонентов требует соответствующего блока питания, поэтому мы использовали Corsair AX860i (www.corsair.com), обеспечивший надежное энергоснабжение, работая эффективно и практически беззвучно; к тому же он имеет модульную структуру, которая уменьшает количество пучков кабелей, а это улучшает вентиляцию. В качестве монитора 4K мы взяли Philips 288P6 (www.philips.co.uk).

только выигрывает от увеличения значков, которое облегчает использование 4K по сравнению с *Xfce*. В остальных областях значки и шрифты малы, что несколько неудобно, но их, по крайней мере, можно без проблем увеличить — или через элементарные настройки ОС, или скачав инструмент *Elementary Tweak*, который позволяет менять размер значков и шрифтов по всей системе, делая elementaryOS более удобной. Чтобы установить *Tweaks*, откройте терминал и введите

```
sudo apt-add-repository ppa:versable/elementary-updat
sudo apt-get update
sudo apt-get install elementary-tweaks
```

Затем вы получите доступ к этим дополнительным настройкам, открыв System settings и выбрав Tweaks. Однако будущая версия elementary OS (имеющая теперь кодовое имя Freya) обещает лучшую поддержку HiDPI.

И, наконец, последней — хотя это не умаляет ее значимости — мы запустили Fedora 20 с ра-

«Рабочие столы добавляют поддержку высокого разрешения.»

бочим столом KDE. Подобно Gnome, KDE поддерживает настройку HiDPI для высокого разрешения, хотя результаты пока не особо успешны. Увеличение рабочего пространства — дело хорошее, и значки рабочего стола остаются удобными и выглядят приятно, но шрифты на Панели задач, как и в Cinnamon, очень мелкие и трудно читаются. Программа запуска приложений Fedora Kickoff работает отлично, и хотя шрифт довольно мелкий, читать его вполне можно. Копание в настройках Fedora обеспечило нас рядом опций по настройке размера значков и шрифтов, делающих рабочий стол KDE приятнее в использовании. В принципе, мы бы сказали, что KDE в своей нынешней форме чуть более дружелюбен к 4K, чем Cinnamon без индивидуальной настройки. Однако на момент

подготовки к печати KDE выпустил Plasma 5.0, новейшую версию KDE. Помимо ряда улучшений производительности и графики, она также вводит «конвергированную оболочку Plasma», разработанную с целью донести рабочий стол и удобный интерфейс пользователя до разнообразных нестандартных устройств — подобно цели Canonical сделать Unity удобной в использовании на смартфонах, планшетах и т.д. Это обновление также содержит поддержку HiDPI, что делает рабочий стол намного лучше в использовании с мониторами 4K.

В общем, мы согласились на том, что рабочий стол Ubuntu Unity предлагает наилучший опыт работы, обеспечивая удобный интерфейс при разрешении 4K без дополнительной настройки, что мы относим к «опыту конвергенции» Canonical, подразумевающему настройку Ubuntu под несколько разных устройств и размеров экрана. Создателям более традиционных сред рабочего стола придется еще потрудиться, особенно если разрешение 4K войдет в норму.

Используете более традиционный рабочий стол? Этот вовсе не значит, что вам придется страдать от крошечных значков и трудноразличимых шрифтов. Есть масса способов настроить интерфейс пользователя вашего дистрибутива для лучшей работы при высоком разрешении — см. врезку внизу. Степень сложности и итоговые результаты будут разными, но идеальное изображение 4K в Linux стоит усилий.

Состояние 4K и Linux

Итак, что же мы выяснили о состоянии поддержки 4K в Linux? Основной вывод такой: поддержка вполне нормальная, и есть еще более многообещающие заделы на будущее. Имея подобное оборудование, вы должны быть в состоянии присоединить монитор 4K к своему компьютеру Linux и заставить его работать при разрешении



► Nvidia GTX Titan Black — одна из самых мощных видеокарт в мире — поэтому мы применили две таких видеокарты для обеспечения отличной производительности 4K.

3840×2160. Однако удобство этого разрешения зависит от дистрибутива и среды рабочего стола, которые вы используете.

Поработав с 4K, мы обнаружили, что без дополнительной настройки удобнее всего работать в Ubuntu с Unity. Другие дистрибутивы и рабочие столы поддерживают 4K с переменным успехом, хотя, приложив некоторые усилия, вы очень скоро добьетесь, чтобы ваш рабочий стол стал комфортен. Внушает надежду то, что многие рабочие столы добавляют поддержку более высокого разрешения, в основном с настройкой HiDPI (High Dots Per Inch) — побочным продуктом для пользователей, желающих заменить OS X на Linux на своих MacBook с дисплеем Retina.

Если вы хотите получить максимально приятную работу с 4K в Linux, то мы бы пока рекомендовали немного подождать. Дополнительная поддержка как в дистрибутивах, так и в программах позволит добиться более удобного взаимодействия вашего компьютера с 4K, а тем временем оборудование с поддержкой 4K продолжит падать в цене. К тому же на данный момент выбор контента для 4K не особо богат. Netflix выразил интерес к вещанию контента 4K, но поскольку Netflix недоступен в Linux, если не заменить плагин Silverlight на Pipelight, вы можете позволить себе немного подождать. И все же, если будущее за 4K, то мы совершенно уверены, что Linux безболезненно станет частью этого будущего. **LXF**

Масштабирование до 4K

Скорейший и простейший способ настроить интерфейс пользователя — использовать ярлык Zoom, который работает как с рядом приложений, так и с некоторыми менеджерами файлов. Просто удерживайте Ctrl, и затем нажмите + или прокрутите колесико мыши. Размер шрифта, а также значки и изображения, которые вы используете, увеличатся. Это быстрый способ облегчить работу с экраном 4K, особенно удобный для таких браузеров, как *Firefox* и *Chromium*, где можно мгновенно изменить размер текста на сайте. В других результатах мы столь впечатляющие — иногда значки или изображения будут казаться несколько размытыми — однако если нельзя даже прочитать напечатанное, скажем, в *Libre Office*, то это выход.

Еще лучше производить настройку с помощью системных настроек вашего дистрибутива.

Изменение размеров значков, шрифтов и DPI шрифтов чрез графический интерфейс намного удобнее для пользователя, а некоторые настройки отображения, такие, как в Ubuntu, на ходу изменяют внешний вид рабочего стола — то есть вы видите свои изменения прямо в процессе их внесения, что сильно облегчает задачу точной настройки вашего дистрибутива под 4K. Другое преимущество — то, что эти настройки повсеместны, то есть ваши изменения будут применяться практически ко всем вашим программам, и вам не придется корректировать каждое приложение по мере их использования.

Используете в качестве среды рабочего стола Gnome 3.10 или выше? Можете включить настройку HiDPI, введя в командную строку:

```
gsettings set org.gnome.desktop.interface scaling-factor 2
```

Это настроит интерфейс пользователя Gnome на большее соответствие высокому разрешению. Если вам это не понравится, можете вернуться к размеру по умолчанию, заново введя тот же самый код, но изменив 2 на 1.

Вы можете также самостоятельно настроить размер шрифта следующей командой:

```
gsettings set org.gnome.desktop.interface text-scaling-factor 0.9
```

Вместо 0.9 введите нужный вам масштаб текста. Изменения отобразятся немедленно.

Для настройки других сред рабочего стола можно прибегнуть к помощи нашего старого проверенного друга *xrandr*, с помощью следующей команды:

```
xrandr --output DFP1 --scale 1.5x1.5
```

Замените 1.5 на нужный вам масштаб.



Arch Linux: Ваш гибкий друг

**Джонни Бидвелл шагает в узкий портал,
открывающий широкие возможности.**

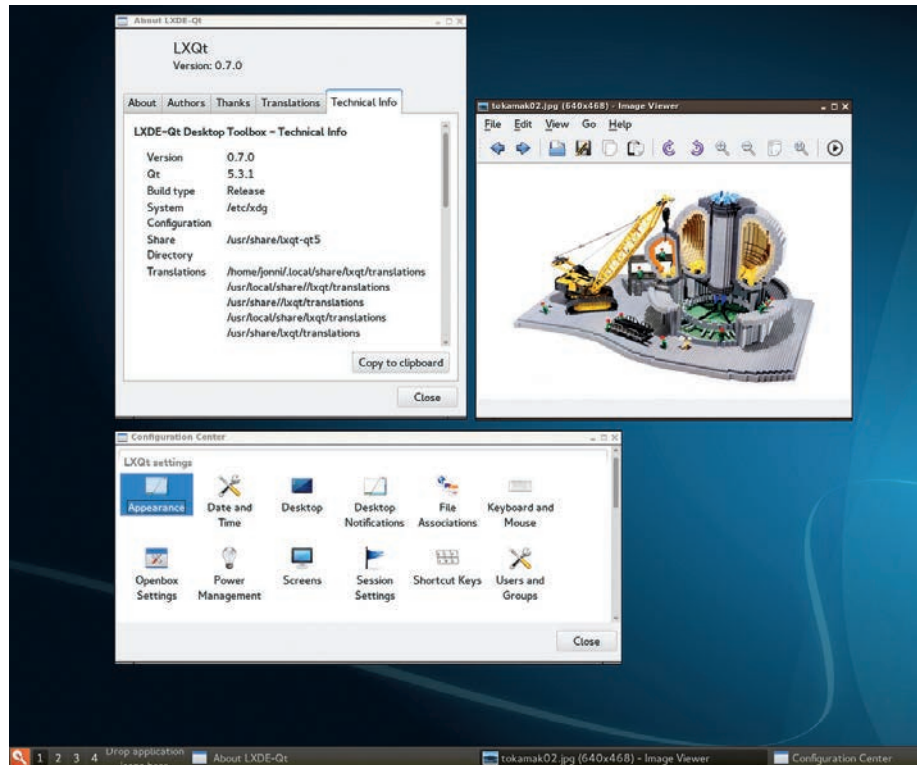
Аrch Linux 0.1 дебютировал в 2002. Его создатель, Джадд Вайнет (Judd Vinet), стремился сделать независимый дистрибутив, простотой подобный Crux, Slackware и BSD. Сейчас Arch Linux находится под руководством Аарона Гриффина (Aaron Griffin) и считается одним из главных игроков на арене дистрибутивов. Среди таких гигантов, как Ubuntu, Mint, Debian, Fedora и openSUSE, Arch

занимает особое место. Несомненно, есть и другие варианты, тоже обещающие сравнимую гибкость (Gentoo, Linux from Scratch), минимализм (Damn Small Linux) и самые актуальные обновления, но ни у одного из них нет столь обширной аудитории. Начиная с минимальной базовой установки, пользователь сам принимает все решения, обустривая систему на свой вкус — или, как говорит Вайнет: «Arch — такой, каким вы его сделаете».

Arch ориентирован на средних или продвинутых пользователей, поскольку без терминала и редактирования файлов вручную здесь не обойтись. Подробная документация есть в wiki Arch, и если вы уже наигрались с Ubuntu или Mint и хотите испытать себя, можете попробовать его на запасной (или виртуальной) машине. Если процесс установки вас не отпугнёт, есть шансы, что вы поладите. Arch стремится представить пользователю всю систему до мельчайших подробностей, но лишь с тем, чтобы внутри она стала логично организованной. Так что, если запастись терпением, можно всерьёз продвинуть свои компьютерные навыки. С другой стороны, если вы только что попробовали Ubuntu и ищите альтернативу по причине встретившихся трудностей, Arch вряд ли решит ваши проблемы. А вот производный от Arch Manjaro Linux, с акцентом на удобство использования, возможно, как раз то, что нужно.

Разборчивые пользователи Arch располагают новейшими версиями графических драйверов (чистых и проприетарных), web-браузеров и практически любых программ, что на ум взбредёт. На форумах Ubuntu это зачастую больной вопрос: репозитории Canonical отстают от релизов разработчиков. Впрочем, на форумах все темы большие, а консервативная стратегия обновления всё же способствует созданию более тщательно протестированных и стабильных дистрибутивов. К слову, стабильные релизы Debian выпускаются раз в три года, так что на момент написания статьи приверженцы Arch наслаждаются новехоньким ядром 3.15, тогда как пользователи текущей стабильной версии Debian (под кодовым именем Wheezy) все ещё довольствуются 3.2. Конечно, эта ветка по-прежнему поддерживается, и все обновления безопасности применяются вовремя. И справедливо будет сказать, что пользователи Debian могут использовать заплатки для получения более новых версий программ... только не тех, что — как Gnome — зависят от многих новых библиотек. Опять же, в Ubuntu есть PPA, в Gentoo — Overlays, и в Fedora есть другие репозитории, такие как RPM Fusion.

На данном этапе Arch Linux существует в двух архитектурах: i686 (32-бит) и x86_64 (64-бит). Это значительно упрощает процесс сопровождения



► Новый рабочий стол LXQT на Arch Linux наверняка ждет популярность.

и тестирования новых пакетов. Когда Arch только вышел (и до появления 64-разрядных процессоров было еще два года), поддерживались только i686. Именно поддержка этой микроархитектуры, применяемой в чипах Pentium Pro, появившихся в 1995-м, отчасти стала причиной его популярности. Другие дистрибутивы все еще сидели на i386, так что Arch воспринимался как более передовое

мультибиблиотечные хранилища — небольшие накладные расходы по поддержке i686 наряду с x86_64 можно считать хотя бы частично оправданными.

Установка Arch

Устанавливается Arch Linux как обычно: скачиваете ISO-образ и создаете загрузочный CD или USB-носитель. Можно выполнить это и из существующей установки Linux, но такой подход не лишен своих подводных камней. Установив новую систему на тот же диск, вы не сможете изменить границы разделов без обращения к загрузочному CD. Кроме того, из другого 32-битного Linux установить 64-битную систему Arch практически невозможно. А сделать наоборот будет, мягко говоря, нетривиально. В общем, если у вас есть CD-дисковод или свободная флэшка, пойдите традиционным путём. ►►

«Arch Linux представляет пользователю систему в мельчайших подробностях.»

решение. Согласно оценкам, i686 сейчас используют менее 10% пользователей, и не исключено, что поддержка этой архитектуры скоро разделит участь динозавров. Но пока не отпала необходимость запускать 32-разрядные двоичные файлы — и, следовательно, сопровождать 32-битные

к релизам разработчиков, и к заплаткам прибегают только в случае крайней необходимости. Например, стандартный пакет ядра обычно содержит лишь несколько исправлений (два, на момент написания), а не целую кучу, как в других дистрибутивах. Arch подчеркивает, что ответственность за управление системой лежит на пользователе, и дает средства делать это эффективно, что не значит просто. Его логика в том, что излишнее упрощение вредит контролю. У Arch очень сильное сообщество, где пользователей поощряют делиться своими проблемами (внятно!) и решениями. Пользователи Arch вольны настроить свои системы под любую прихоть: будь то удаленный сервер, полноценный рабочий стол KDE или аудиостанция в режиме реального времени, а адаптироваться Arch сумеет. Проприетарное и другое лицензионное программное обеспечение (например, Flash и кодеки MP3) доступны в репозиториях, поскольку функциональность здесь превыше идеологии.

Философия Arch

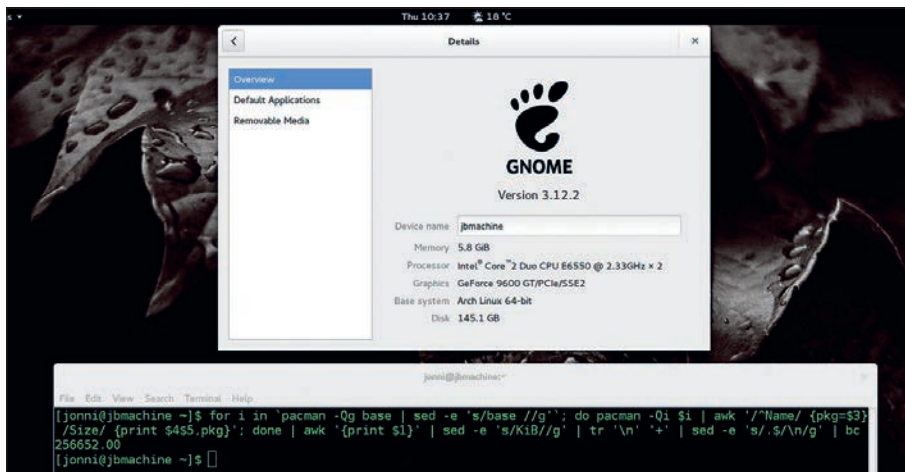
Философия Arch Linux базируется на пяти принципах: простота, приоритет корректности кода над удобством, ориентированность на пользователя, открытость и свобода. Он призван обеспечить исходный фундамент, на котором пользователь строит что захочет. Linux — штука сложная, иногда и запутанная, и Arch не пытается спрятать это за слоями абстракции или GUI, а, напротив, старается представить как можно стройнее и прозрачнее. Программное обеспечение максимально приближено

Если вы ожидаете увидеть шикарный графический установщик, который настроит всё за вас, задав пару простых вопросов, и шустрый индикатор выполнения, то будете разочарованы. Установщик просто подкинет вам строку `zsh`, чтобы вы настроили свои диски, сетевые соединения и локализацию. В установщике есть всё необходимое для создания диска с GPT-разделами, но вы можете предпочесть `Gparted` на live CD, потому что с `cgdisk`, `gdisk` и `parted` бывает немало мучений. Все это более чем полно изложено в официальном руководстве по установке для начинающих (<http://bit.ly/BeginArch>), и здесь мы представим только беглый обзор.

Образы самых коварных беспроводных прошивок включены в это live-окружение, дабы можно было (обойдя ситуацию Уловки-22 из-за их отсутствия) заставить работать свою карту беспроводной сети. Все настроив, вы можете загрузить и установить (весьма) аскетичную систему в нужный раздел, с помощью скрипта `pacstrap`, и сгенерировать файл `fstab`. Затем вы сможете попасть в свою новую установку через `chroot` и (о радость!) выполнять практически те же настройки, что и в live-среде, а также кое-что ещё. Хорошо бы на данном этапе настроить постоянное беспроводное соединение. Для этого установите пакет `linux-firmware`, или любой другой, который требуется для вашей беспроводной карты, и добавьте нужный модуль ядра в файл `.conf` в `/etc/modules-load.d/`.

Для управления сетевыми соединениями в Arch есть прекрасный инструмент под названием `netctl`. Он бодро управляет всем, от простых статических IP-соединений Ethernet до VPN и беспроводных сетей с паролями. Независимо от вашего варианта установки, вам нужно создать профиль в `/etc/netctl`; скорее всего, найдётся пример с похожими требованиями, так что это почти безболезненно. Для простых WPA(2) сетей, вы можете даже использовать инструмент `wifi-menu`. Для пользователей ноутбуков (или тех, кто часто переключается между сетями), есть сервисы `netctl-ifplugd` (для проводной) или `netctl-auto` (для беспроводной) для быстрого подключения к известным сетям.

Наконец, нужно задать пароль для root и установить загрузчик, которых здесь несколько. Как правило, все выбирают `Grub`, но в любом случае, чётко следуйте инструкциям: кому нужна



➤ Строка подсказки сообщает, что базовая установка занимает всего 256 МБ. Плюс, у нас есть новый гном.

незагружаемая система? Обратите особое внимание на дополнительные инструкции для материнских плат с UEFI. И вот, у вас есть (как мы надеемся) полностью функционирующая система Arch. Но что же она умеет делать? Терпение, дорогой читатель, терпение.

Все официальные пакеты практически не отстают от релизов разработчиков, но прошли тщательное тестирование, так что стабильность не страдает. Arch является примером модели так называемых «плавающих релизов», поэто-

можно впихнуть его через сторонние репозитории (или перейти на Gentoo), но многие пользователи (и все нормальные люди) избегают подобных сомнительных практик. В Arch, напротив, новый релиз был тут же упакован и ждал скачивания на низком старте.

Нон-стоп

Модель плавающих релизов работает отчасти благодаря элегантной и простой структуре Arch. Она позволила разработчикам с гармоничной синхронностью перевести всех с `SysVinit` на `Systemd` и спрятать цифровые подписи в `Pacman` во мгновение ока. Пакет `filesystem` содержит базовый каталог и основные файлы настройки в `/etc`, поэтому, внося изменения в этот пакет, вы применяете

новый формат. Так, в 2013-м постановили, что каталоги `/bin`, `/sbin` и `/usr/sbin` излишни и хранящиеся там бинарники следует поместить в `/usr/sbin`. Подобные низкоуровневые подвижки невозможны при традиционной модели релиза, где конфигурации определяются заранее и не меняются, пока не начинает задумываться новый. Но кое-кто от перехода на `/usr/sbin` пострадал — в первую очередь те, кто использует неофициальные пакеты. На домашней странице была размещена справочная информация (архив находится на <http://bit.ly/1i7jWqE>), но к сожалению, многие проигнорировали и её, и предупреждения `Pacman`. И потом лили слёзы. Без уведомлений о необходимых изменениях порой не обойтись, но они, как правило, касаются только отдельных компонентов, а не глобальных изменений системы. В любом случае, если обновление потенциально способно навредить вашей системе, его надо слить, не приводя в действие, и в этом пункте следует навести свои органы зрения на www.archlinux.org, не то придётся вам рвать и метать. Разрушения можно вчинить только ручным принуждением.

По мере добавления новых версий пакетов в репозитории, другие, зависящие от них, следует обновить или по крайней мере пересобрать, чтобы избежать поломки. Команда Arch делает неоценную работу по поддержанию согласованности

«Официальные пакеты не отстают от релизов разработчиков.»

му за очень редкими исключениями, при условии, что вы регулярно выполняете `pacman -Syu`, ваша установка всегда будет в актуальном состоянии. Дистрибутив как таковой не обновляется ни каждые 6 месяцев, ни через 3 года, ни за период максимальной солнечной активности (11-летний цикл, кстати). ISO-релизы — это просто копии текущих базовых пакетов. Когда в марте вышел Gnome 3.12, он получил положительные отзывы (за счет исправления многих недостатков серии 3.x), но не смог попасть в консервативные циклы релизов других основных дистрибутивов. За что его и успели окрестить «бездомный рабочий стол». Конечно, в некоторых дистрибутивах

Маломощные системы

Скорее всего, ваша установка заняла менее 1 Гб на диске. Вероятно, придётся пожертвовать немного больше, чтобы расширить свои возможности, но если пространство ограничено, хватит и этого. Arch популярен среди владельцев более старого и менее мощного оборудования, во многом благодаря своему минимализму. Обратитесь к статье о низкоресурсных приложениях и окружениях рабочего стола из августовского номера [см.

LXF186, стр. 46]. Благодаря проекту Arch на ARM (<http://archlinuxarm.org>), вы можете установить его и на свой Raspberry Pi — образ легко умещается на SD-карту в 2 Гб. Но Arch также и достойный кандидат для установки на более мощные машины. Стандартное ядро Arch содержит последнюю стабильную версию с <http://kernel.org>, так что вы всегда будете наслаждаться новейшими средствами аппаратной поддержки.

всех официальных пакетов, чтобы совместимость не пострадала от несостыковки версий, но в конечном счете безопасность обновлений зависит от пользователя. Следовательно, вносить в систему частичные обновления крайне нежелательно: это может ее угробить. Обновляйте или всё (с **pacman -Syu**), или ничего. Новые пакеты, наряду с теми, что на них построены, представлены в репозитории **testing**. И хотя новичков прельщает возможность урвать новейшее ПО, делать это далеко не благоразумно. Ведь в **testing** программы находятся ради тестирования возможных сбоев. И в этом случае с обломками разбираться вам.

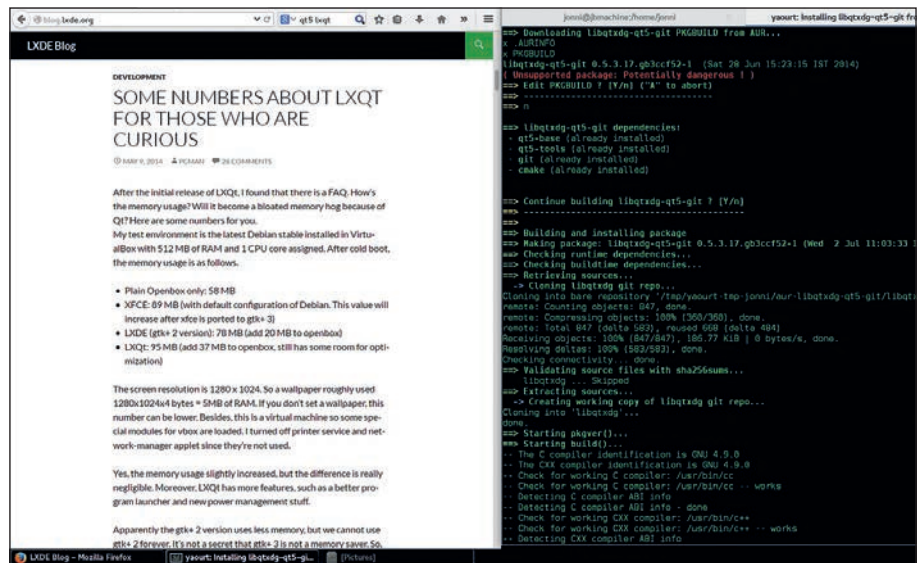
man pacman

Менеджер пакетов Arch называется *Pacman*. Легкий, быстрый и простой, он воплощает философию Arch. А всю тяжелую работу выполняет внутренняя библиотека под названием *libalpm*. *Pacman* работает с официальными или неофициальными репозиториями, которые хранят пакеты в формате **.pkg.tar.xz**. Это сжатые архивы LZMA2, содержащие файлы пакетов и каталогов, а также файлы с метаданными пакетов, контрольными суммами, скриптами пост- (де)инсталляции и т.п. По умолчанию в Arch используются три репозитория:

- » **Core** Базовая система, которую вы получаете при установке.
- » **Extra** Другие официально поддерживаемые пакеты.
- » **Community** Содержит пакеты, одобренные сообществом Arch. Именно «Доверенные пользователи», а не официальные разработчики Arch занимаются поддержкой пакетов, но это не значит, что нельзя создавать свои собственные.

Помимо работы с отдельными пакетами, *Pacman* поддерживает и группы пакетов. Если вам необходимо сделать компиляцию, понадобится группа **base-devel**. В неё входят *gcc*, *make* и прочие утилиты, которые творят свои чудеса. Рабочие столы Gnome, Mate, LXDE и KDE также составляют группу, и можно выбрать, какие части этих гигантов установить. Группа KDE — сама крупная, в ней более 200 пакетов, но, к счастью, есть ещё одна, называемая *kde-meta*, где они разбиты на 16 подгрупп, это облегчает процесс.

При обновлении пакетов появляются новые файлы настройки, которые могут конфликтовать



» Yaourt упрощает нудный процесс сборки и упаковки компонентов LXQt.

или не конфликтовать с предыдущими: иногда пользователи вносят свои изменения или меняются установки по умолчанию. *Pacman* проверяет хэш md5sum текущего файла, чтобы это обнаружить. Если изменений нет, файл перезаписывается новым. В противном случае *Pacman* устанавливает новый файл настройки с расширением **.pacnew**. Пользователь получает соответствующее сообщение и может принять решение сам, желательно сразу по завершении обновления. Можно

«Вносить в систему частичные изменения крайне нежелательно.»

проверить различия, используя стандартные инструменты: например, если пакет *openssh* включает новый **ssh_d_config** —

```
$ diff /etc/ssh/ssh_d_config{,.pacnew}
```

Обратите внимание на раскрытие команды в *Bash*: это чтобы не нажимать лишних клавиш. Пользователю необходимо включить в новый файл все новые настройки вместе с изменениями. Если последних не очень много, то это проще всего сделать, отредактировав файл **.pacnew** и перезаписав оригинал.

На самом деле, *makepkg* и *Pacman* являются компонентами небезызвестной ABS, или Arch Build System. Сюда также входит дерево ABS — это иерархия, содержащая PKGBUILD для всех официальных пакетов Arch. Установив пакет *abs* и выполнив одноименную команду, вы получите копию этого дерева в **/var/abs**. Это, а так же пакет *base-devel*, поможет вам менять официальные пакеты на своё усмотрение: включая/отключая дополнительные функции, выбирая конкретные версии программного обеспечения или настройки CFLAGS. Что касается последнего, то благодаря сценарию *pacbuilder* можно перекомпилировать всю систему с **-O3**. Столь отважное поведение — не привилегия одних только пользователей Gento! Поддержка 386 процессоров была исключена из ядра в версии 3.8 («Скатертью дорога», сказал Линус). Но если вы все еще горюете, что Arch не будет работать на вашей 386, можете употребить *pacbuilder*, старое ядро, свою богатую изобретательность и море свободного времени на исправление ситуации. На этом мы завершаем обзор этого прославленного дистрибутива. И да будут ваши скитания по Арчианским равнинам плодотворными и приятными, ваша система — быстрой, а файлы настройки — свежими. **LXF**

Собери пакет сам

Пакет исходников может собрать любой (двоичные файлы не допускаются, по очевидным соображениям безопасности) — и добавить его в Arch User Repository на радость сообществу. Это может быть ваше собственное ПО или чужое; возможно, новинка или отсылка в git; и, возможно, при наличии дополнительных функций или без (дабы избежать проблем с зависимостями).

Пакеты могут состоять из одного файла, именуемого PKGBUILD, содержащего подробности

исходников, контрольные суммы и любые необходимые заплатки, вместе с функциями компиляции и установок. Выполнив команду **makepkg** с параметром **-S** для готового PKGBUILD, вы сделаете его доступным для размещения в AUR. Без **-S** получится двоичный пакет, устанавливаемый по **pacman -U**.

Пользователи оценивают пакеты в AUR, популярные (и приемлемые) одобряются доверенным пользователем и добавляются в репозиторий сообщества. При установке AUR- (или любых других

несанкционированных) пакетов следует проявлять осторожность, проверяя PKGBUILD и заплатки, чтобы не повредить систему. Однако, если у вас есть опасения, можно поступить проще и использовать инструмент-помощник AUR Helper, например, Yaourt (Yet Another User Repository Tool). Это оболочка для *Pacman* (с тем же синтаксисом), позволяющая автоматизировать процесс сборки и установки AUR-пакетов. Кроме того, у него приятный и красочный вывод.



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.

Проклятая ложь и руководства

Надеюсь, вы согласитесь со мной в том, что учебники в Linux Format отличные. Я сам пишу некоторые из них! Но не казалось ли вам, что в них все идет слишком гладко? Не бывает ошибок аутентификации, и не появляются непонятных сообщений об ошибках... А замечали ли вы, что во всех видеоруководствах все тоже «просто работает»?.. Что читатели думают об этом? Представляют ли они некоего гуру, который никогда не оступается? Или им понятно, что было на самом деле — что автор две недели бился, заставляя это наконец заработать, и однажды даже всерьез подумал, заметит ли редактор, если вместо этого отправить ему статью про оригами? [Я заметил, — ред.] Понимают ли они, что явная склейка в видеоролике означает момент, когда рассказчик столкнулся с такими трудностями, что ему пришлось вернуться к предыдущему снимку виртуальной машины, сделанному накануне утром?

Меня беспокоит, что кто-то пугается или чувствует себя некомфортно, когда его личный опыт расходится с идеализированным миром урока. Жизнь устроена так (и я часто повторяю это озадаченным студентам), что гораздо больше способов сделать что-то неправильно, чем правильно. И если это вас ободрит, скажу не тая, что у меня все постоянно идет не так. Смешной пример неудачного демо приведен в блоге Роба Розна [Rob Rohan] (<http://bit.ly/perlpalaver>): незадачливый программист на Perl хочет ввести программу через утилиту распознавания речи. chris.linuxformat@gmail.com

Чисто поверхностно

В Active Directory много чего имеется; например, в ней можно хранить массу всяких данных пользователя — адрес электронной почты, дату окончания срока действия пароля и т.д. Также есть расширения схемы LDAP (определенной в RFC 2307),

Эзотерическое системное администрирование из причудливых заворотов кишок серверной



NetworkManager

Утилита, которую все мы принимаем как данность — *NetworkManager* из Red Hat — только что перенесла подтяжку лица.

Если на вашем ноутбуке стоит Linux, на рабочем столе скорее всего где-то найдется маленькая иконка, принадлежащая апплету *nm-applet*. Он предоставляет доступ к сервису, который управляет сетевыми подключениями. Он называется (без сюрпризов...) *NetworkManager*, и его миссия состоит в том, чтобы сеть «просто работала» с минимальным вмешательством пользователя или вообще без него.

Для этого *NetworkManager* автоматически активирует сетевые интерфейсы, подключается к сетям Wi-Fi, задает IP-адреса и настраивает маршрутизацию «так, как считает нужным». Информация о состоянии подключения передается по D-BUS любому заинтересованному приложению, которое позволяет (например) почтовому клиенту мигмом узнать, подключен ли компьютер к Интернету.

Признаю, что в прошлом я был не слишком добр к этой утилите, недовольно ворча, когда она ни с того ни с сего перезаписывала */etc/resolv.conf*, огорчаясь, когда она молча переключалась

с VPN на незашифрованное соединение, и теряясь в догадках, запущена ли она вообще. Но только что вышла новая версия (0.9.10), и я подумал, что как минимум честно было бы вывести ее на сцену и дать ей право раскланяться под аплодисменты.

Последняя версия еще не попала в репозитории; если вы хотите ее попробовать, придется скомпилировать ее из исходников (скачайте *NetworkManager-0.9.9.98.tar.xz* от gnome.org с <http://bit.ly/1mg14Qb>).

У программы длинный список улучшений, включая поддержку Data Center Bridging, шаг в сторону модульной архитектуры на основе плагинов и даже возможность подавить эти надоедливые обновления *resolv.conf*. Также есть новая утилита настройки *nmtui* на базе *curses* (не путать с утилитой *systemconfig-network* в Red Hat).

У командной утилиты *nmcli* масса улучшений, хотя я поклялся бы, что большинство из вас вообще не знали о ее существовании! Теперь по шине D-BUS передается больше информации о состоянии соединения, а новая утилита более аккуратно считывает (и не пытается изменить) все существующие настройки активных интерфейсов.

Если вы никогда не пользовались этой командой, попробуйте. Например, вот так можно вывести список всех доступных точек доступа Wi-Fi:

```
$ nmcli device wifi list
```

а так — вывести список текущих подключений:

```
$ nmcli connection list
```

а так — получить общую информацию:

```
$ nm-tool
```


Интеграция Linux и Windows

Добрый Доктор показывает, как создать контроллер домена Active Directory, скомпилировав *Samba 4* из исходников.

В нескольких последних статьях я рассказываю о *Samba* — сегодня часть третья — и о том, как с ее помощью интегрировать Linux в сетевое окружение Windows. Мы начали с простого файлового сервера, а затем посмотрели, как аутентифицировать пользователей в Linux, используя учетные записи из Active Directory.

Моя миссия в этом месяце — показать, как создать контроллер домена Active Directory на основе *Samba 4*. Microsoft также называет этот контроллер «Сервисами домена Active Directory (Active Directory Domain Services — AD DS)». Это одна из более сложных задач, за которые я брался на своих уроках, и она более директивна («сделайте это, затем это, потом это...»), чем обычно. Но я надеюсь, что у вас все получится! Рветесь в бой? Хорошо. Прежде чем начать, нужно решить, как будет называться домен. Я выбрал «LXF.LOCAL». Это имя (или его вариации) фигурирует на нескольких этапах установки.

Установим Ubuntu Server

Итак, если вы хотите следовать за мной, начните с загрузки и установки ISO-образа сервера Ubuntu 14.04 на настоящий компьютер (если есть под рукой) или на виртуальную машину. Загрузить его можно с <http://releases.ubuntu.com/14.04>. Окна установщика не должны вызывать особых сложностей — я выполнил базовую установку с файловой системой ext4 и без логических разделов и задал имя хоста lxfserver. (Примечание: имя хоста должно отличаться от имени домена, поэтому мы не можем назвать хост просто lxf.)

В окне выбора программ я указал сервер SSH, чтобы впоследствии иметь возможность подключиться к этому компьютеру удаленно. Других серверных компонентов я не выбирал, так как мы планируем устанавливать *Samba 4* из исходников. При желании, наверное, можно воспользоваться и более ранней версией Ubuntu, например, 13.10, или Debian 7 (Wheezy).

Как вы, должно быть, знаете, прямой вход в систему от имени root в Ubuntu по умолчанию отключен, и вам придется предварять каждую команду, требующую прав root, префиксом **sudo**. Этих прав требуют почти все команды, которые мы будем запускать на этом уроке, а от постоянного набора **sudo** можно рехнуться, поэтому после входа в систему советую переключиться в оболочку root командой

```
$ sudo -s
```

Задаем статический IP-адрес

Установив базовый сервер, задайте ему фиксированный IP-адрес. Для своего сервера я выбрал адрес **192.168.1.100**. Вам нужно убедиться, что выбранный адрес (а) находится в вашей сети и (б) не будет выделен другим компьютерам. Чтобы задать IP-адрес, откройте файл **/etc/network/interfaces** и измените запись для интерфейса eth0 на следующую:

```
auto eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.254
dns-nameservers 192.168.1.100 8.8.8.8
```

```
dns-search lxf.local
```

Надеюсь, большая часть этих параметров не требует пояснений. Параметр gateway (192.168.1.254) — это шлюз вашей сети по умолчанию. В моей домашней сети это внутренний IP-адрес моего широкополосного роутера. Первый сервер в dns-nameservers — это сам сервер (да, мы запустим на нем DNS), а второй — сервер имен Google, который мы указали на всякий случай. Есть вариант и получше — указать здесь DNS-сервер своей интранет-сети или DNS-сервер, предоставленный провайдером. В параметре **dns-search** указан наш домен (в нижнем регистре). При запуске интерфейса эти настройки DNS передадутся в **/etc/resolv.conf**, обычный файл настройки DNS. (Чтобы узнать о нем подробнее, загляните на map-страницу для resolvconf.)

Теперь нужно добавить в файл **/etc/hosts** строку для разрешения имени сервера. В моем случае она выглядит так (не забудьте изменить IP-адрес и имя хоста в соответствии со своими):

```
192.168.1.100 lxfserver.lxf.local
```

Последняя настройка сети — смена имени хоста в **/etc/hostname** на

```
lxfserver.lxf.local
```

и тут, стыдно сказать, я решил, что простейший способ ввести все эти изменения в силу — просто перезагрузиться:

```
# shutdown -r now
```

После загрузки системы проверьте, что IP-адрес соответствует заданному ранее, командой

```
# ifconfig eth0
```

и загляните в **/etc/resolv.conf** для проверки, что DNS-серверы заданы верно.

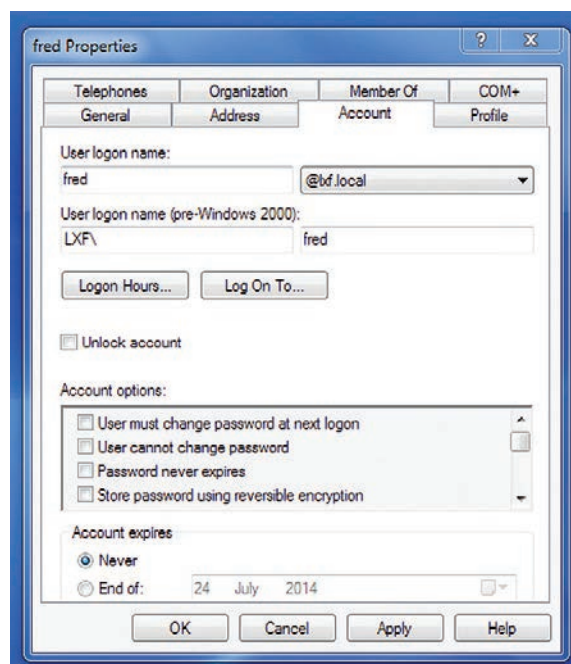
Затем последуем общепринятым рекомендациям и обновим систему:

```
# apt-get update
```

```
# apt-get upgrade y
```

На это может уйти несколько минут.

»



» Мы видим пользователя, которого я добавил (см. стр. 53) к своему домену, в утилите настройки удаленного сервера в Windows 7.

Установка необходимых пакетов

Теперь скомпилируем *Samba 4* из исходников. Нужно установить гору пакетов, так что обычная шутка о приготовлении чашечки чая здесь не пройдет... может быть, чайничек чая? В следующем списке *git* нужен для загрузки исходников *Samba* из репозитория *Git*, а *ntp* (Network Time Protocol) — чтобы время на сервере оставалось точным. Метапакет *build-essential* содержит утилиты компилятора. Большинство пакетов содержат библиотеки и заголовочные файлы, необходимые для их связи друг с другом. Команда приведена ниже — я разбил ее на несколько строк для удобства чтения, но вы можете набрать ее в одну строку, опустив обратные слэши:

```
sudo apt-get install git build-essential \
libacl1-dev libattr1-dev \
libblkid-dev libgnutls-dev \
libreadline-dev python-dev \
python-dnspython gdb \
pkg-config libpopt-dev \
libldap2-dev dnsutils \
libbsd-dev attr \
krb5-user docbook-xsl \
libcups2-dev libpam0g-dev \
ntp -y
```

Во время установки система настройки пакетов *debconf* задаст вам несколько вопросов о Kerberos. Задайте «область действия» Kerberos по умолчанию в *LXF.LOCAL*, и задайте имя хоста для сервера Kerberos и административного сервера в *lxfserv* (или в том имени, которое вы выбрали).

Наша следующая задача — загрузить исходный код *Samba* из репозитория *git*:

```
# git clone -b v4-1-stable git://git.samba.org/samba.git samba4
```

Эта команда загрузит последнюю версию *Samba* (в моем случае, 4.1.8) в каталог **samba4**. У меня это заняло минут пять. Для компиляции *Samba 4* из исходников используется стандартная комбинация из трех... хм... команд:

```
# cd samba4
# ./configure --enable-debug --enable-selftest
# make
# make install
```

На моем компьютере это заняло около 25 минут.

Загрузка и сборка Samba

Чтобы проверить свою сборку, оставайтесь в этом же каталоге и скомандуйте

```
# make quicktest
```

Эта команда прогнала серию из более чем 2000 тестов, заняв около 20 минут. Так как в скрипте **configure** мы приняли предложенные по умолчанию пути установки, установленная *Samba* попадет в каталог **/usr/local/samba**. В частности, двоичные файлы будут в каталогах **/usr/local/samba/sbin** и **/usr/local/samba/bin**. Возможно, будет удобнее добавить эти каталоги в переменную окружения *PATH*. Для этого просто припишите строку

```
PATH=$PATH:/usr/local/samba/sbin:/usr/local/samba/bin
```

в файле **/root/.profile**. Если вы не хотите этого делать, всегда остается возможность использовать в командах полные пути.

Одно из преимуществ сборки программы из исходников в том, что можно настроить, как именно будет собираться *Samba 4*. Скрипт **configure** имеет массу параметров для включения или отключения тех или иных возможностей и для выбора каталогов, где будут установлены компоненты программы. Чтобы увидеть все возможные параметры, выполните команду

```
./configure --help
```

В частности, вы можете предпочесть более «традиционные» каталоги для установки *Samba 4* (чтобы двоичные файлы оказались в каталогах, которые уже есть в путях поиска), указав при запуске скрипта **configure** параметры **--prefix**, **--bindir** и **--sbindir**.

Развернем домен

Установив *Samba*, мы можем перейти к развертыванию домена. Для этого воспользуемся *samba-tool* (я предполагаю, что вы скорректировали свои пути поиска, как сказано выше.) Боюсь, команда опять получилась длинной, поэтому набирайте ее внимательно и не забудьте изменить область действия и домен на свои.

```
# samba-tool domain provision \
--realm=lx.local --domain=LXF \
--adminpass=18June2014 --server-role=dc \
--dns-backend=SAMBA_INTERNAL
```

Пароль администратора, который задается в этой команде, очень важен. Не забудьте его! Также помните, что пароль проверяется на надежность, и если он будет слишком простым, команда завершится неудачно.

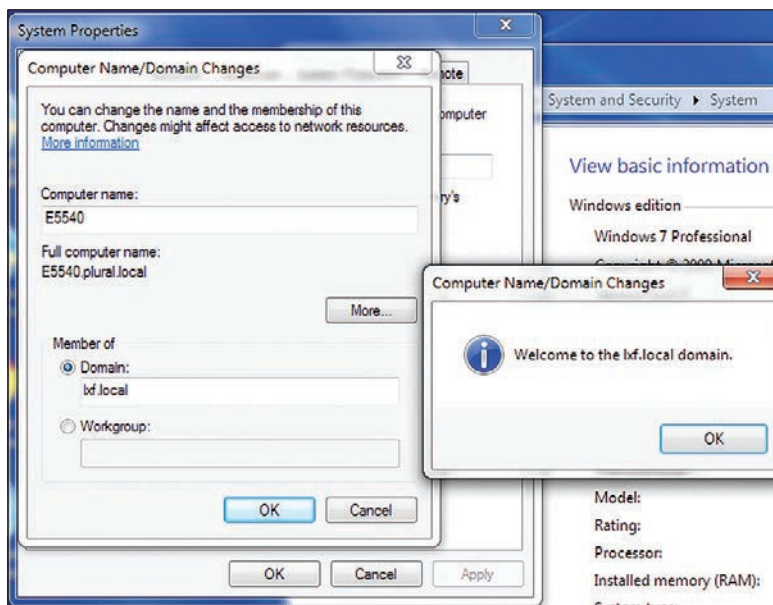
При успешном завершении команды (на моем компьютере на это потребовалось около минуты) вы должны увидеть описание созданного домена, которое выглядит примерно так:

```
Server Role: active directory domain controller
Hostname: lxfserv
NetBIOS Domain: LXF
DNS Domain: lx.local
DOMAIN SID: S-1-5-21-1631141774-1776331329-1621015434
```

Для получения справки по этой команде и просмотра множества других параметров наберите

```
# samba-tool domain provision --help
```

Прежде чем двигаться дальше, выполните команду **date** и убедитесь, что дата и время установлены правильно (и поправьте, если нет), и проверьте, что демон *ntp* запущен. Я кратко



► Пора отпраздновать успех! Мой ноутбук с Windows 7 успешно присоединился к новому домену.

Исходники или бинарник?

Процедуру можно упростить, установив готовые пакеты из репозитория Ubuntu. Например, скачав скрипты загрузки для *Upstart*, вы сэкономите 20 минут на компиляции. Почему же я устанавливал *Samba* из исходников? Да чтобы убедиться, что

у меня последняя стабильная версия *Samba* (на момент написания статьи — 4.1.8, а в репозиториях Ubuntu есть только 4.1.6); чтобы лучше управлять процессом сборки; но главным образом — ну вот так мне захотелось!

описывал эти шаги месяц назад. Синхронизация времени важна, поскольку в Kerberos используются сообщения с временными метками.

Запуск демона Samba

Один из недостатков установки *Samba 4* из исходников — то, что он автоматически в *Upstart* не включается, и его придется запускать вручную после каждой загрузки. Но после сложных команд, о которых я вам рассказывал, это легче легкого:

```
# samba
```

Добавить задание для автоматического запуска в *Upstart* нетрудно, но сейчас я этого делать не буду. У нас есть цели поинтереснее. Дайте демону несколько секунд на запуск, а затем посмотрите порты, которые он слушает. Вы получите представление о том, насколько разнообразные задачи на самом деле решает сервер:

```
# lsof -i | grep '^samba.*IPv4'
```

Список будет длинным — с сервисом имен *netbios*, *ldap* и *secure ldap*, *kerberos* и *DNS*.

Проверка работы Samba и Kerberos

Пора немного присмотреться к нашему домену. Для начала выведем список общих ресурсов (я его малость подсократил):

```
# smbclient -L localhost -U%
```

```
Domain=[LXF] OS=[Unix] Server=[Samba 4.1.8]
```

```
Sharename Type
```

```
-----
```

```
netlogon Disk
```

```
sysvol Disk
```

```
IPC$ IPC
```

Теперь выведем список содержимого ресурса. Для этого нужно аутентифицироваться у сервера *Samba*:

```
# smbclient //localhost/netlogon -UAdministrator%18June2014
```

```
-c ls
```

```
Domain=[LXF] OS=[Unix] Server=[Samba 4.1.8]
```

```
. D 0 Mon Jun 23 14:29:56 2014
```

```
.. D 0 Mon Jun 23 14:30:10 2014
```

Да, информации здесь немного, но мы, по крайней мере, убедились, что можем аутентифицироваться у сервера. Теперь проверим работу *Kerberos*, убедившись, что можем получить от него «билет, дающий право на получение билета»:

```
# kinit Administrator@LXF.LOCAL
```

```
Password for Administrator@LXF.LOCAL:
```

```
Warning: Your password will expire in 41 days on Mon 04 Aug 2014 14:30:07 BST
```

Вроде бы все в порядке. Убедимся, что и вправду получили билет:

```
root@lxfserver:~# klist -e
```

```
Ticket cache: FILE:/tmp/krb5cc_0
```

```
Default principal: Administrator@LXF.LOCAL
```

```
Valid starting Expires Service principal
```

```
23/06/14 15:37:54 24/06/14 01:37:54 krbtgt/LXF.LOCAL@LXF.
```

```
LOCAL
```

Заглянув в урок из прошлого номера [стр. 60, **LXF187**], вы вспомните, что мы уже проводили с *Kerberos* этот простой тест. Разница в том, что в прошлом месяце *Kerberos* был запущен в *Windows* (в составе *Active Directory*), а на сей раз — в *Linux* (в составе *Samba*). Для просмотра и администрирования домена из командной строки *Linux* мы пользуемся *samba-tool*.

Например, можно добавить в домен пользователя:

```
# samba-tool user add fred 'VamBam!'
```

```
User 'fred' created successfully
```

Здесь *fred* — имя пользователя, а *VamBam!* — пароль. Подробное описание процесса добавления пользователей и список параметров команды можно получить, набрав

```
# samba-tool user add --help
```

А получить список пользователей вы сможете командой

```
# samba-tool user list
```

```
Administrator
```

```
krbtgt
```

```
Guest
```

```
fred
```

Подключаем к домену компьютер с Windows

Конечно, чтобы проверить все по-настоящему, нужно подключить к домену *Windows*, и я приведу краткие инструкции о том, как сделать это (в моем случае) для ноутбука с *Windows 7 Professional*.

Сначала следует задать сервер *DNS* ноутбука. Зайдите в *Control Panel > Network and Internet > View Network Status and Tasks > Change Adaptor Settings > [выберите свой сетевой интерфейс] > Properties > TCP/IP V4 > Properties [Панель управления > Сеть и Интернет > Просмотр состояния сети и задач > Изменить настройки адаптера > [выберите свой сетевой интерфейс] > Свойства > TCP/IP V4 > Свойства]*, и затем задайте *192.168.1.100* (или выбранный вами адрес сервера *Samba*) в поле адреса предпочтительного *DNS*-сервера. Не сделав этого, вы не сможете присоединиться к домену.

Чтобы присоединиться к домену, в меню *Пуск* выберите *Computer > Properties > Advanced System Settings > Computer Name > Change > Member of Domain [Компьютер > Свойства > Дополнительные параметры системы > Имя компьютера > Изменить > Является членом домена]* и введите имя домена (в моем случае, **lxf.local**). Если все пройдет удачно, вы увидите сообщение о приглашении в домен (на рисунке, стр. 52), но чтобы изменения вступили в силу, придется перезагрузиться.

Теперь вы должны подключиться к домену *LXF*, аутентифицировавшись у контроллера домена либо от имени пользователя *Administrator* с паролем *18June2014*, либо от имени пользователя *fred* с паролем *VamBam!*. Если получилось, то все отлично! Вы успешно создали контроллер домена *Active Directory* на *Samba*.

Хотя домен можно администрировать из командной строки *Linux* с помощью *samba-tool*, большинство администраторов скорее всего предпочтут установить на компьютер с *Windows* утилиты для автоматического администрирования удаленного сервера (*Remote Server Administration Tools*). Их можно бесплатно загрузить с сайта *Microsoft* (<http://bit.ly/RSATWin7>). Это графические утилиты для просмотра и управления пользователями, компьютерами, групповыми политиками и другими параметрами домена. Загрузив и установив утилиты, их нужно включить, зайдя в *Control Panel > Programs and Features > Turn Windows Features on and off [Панель управления > Программы и компоненты > Включение и отключение компонентов Windows]*.

Готово. Поздравляю с завершением моего высокоскоростного руководства! С нуля до контроллера домена *Active Directory* — мы уложились в три страницы. **LXF**

Благодарности

Должен признаться, что заставить все это работать мне очень помогли два источника. Первый — книга «Разворачиваем *Samba 4*» Марсело Леаля [Marcelo Leal] (доступна в электронном виде на <http://bit.ly/ImpSamba4>), описывающая процесс установки и настройки *Samba* на удивление

подробно. Второй — видеоруководство Дэвида Гогена [David Goguen] на ту же тему (<http://youtu.be/Rf7Hk8qWt1Q>). И в книге, и в руководстве довольно мало информации по теоретической основе, но они реально полезны при развертывании *Samba*.

LINUX FORMAT

Подписывайтесь и читайте Linux Format на iPad или iPhone!

Доступно
в AppStore



А если у вас Android, подпишитесь
на Linux Format через Zinio!

 zinio™
Доступно в Google Play



Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Курс для новичка

Машина должна работать,
а человек — думать.
Витус Вагнер

Время от времени возникает непростая проблема: «Куда послать новичка?» В смысле, что посоветовать ему почитать, чтобы, с одной стороны, это заняло не месяцы и не недели, но дни, а с другой стороны, он бы включился в работу в Linux-окружении хотя бы в половину потенциальной силы, попутно читая ману и быстро набирая опыт. То есть всяческие популярные рассказы, какую менюшку в LibreOffice нужно дернуть, чтобы выровнять строчку по центру, тут совершенно не помогают.

Длительное время для меня таким местом был «Курс „Использование Unix/Linux“ для сотрудников ИЯФ» от Димы Болховитянова, но курс был составлен 15 лет назад и с тех пор не менялся, что привело к его неминуемому устарению. Хотя и сейчас отдельные его части я рекомендую для быстрого просмотра.

Я уж почти отчаялся, но недавно Linux Foundation выложила на системе массовых открытых интернет-курсов edX достаточно качественный проект «Welcome to LFS101x: Introduction to Linux». Знание английского языка безусловно нужно, но если оно имеется, то уже есть куда посылать, и это весьма немало.

И да, кроме самого курса, в аннотациях к разделам мелькают головы вполне себе ярких в мире Linux личностей. Родоначалник тоже там есть — правда, только в самом начале.

E.M.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Работать в терминале 56

Нейл Ботвик управляет учетными записями, создавая пользователей, задает пароли и осваивает разрешения и права владения.



Ругать Luchee 58

Установим с **Маянком Шармой** сервер фото — и давай делиться фотками с друзьями и родными; и даже употребим Raspberry Pi как фоторамку.



Употреблять Systemd 62

Отдельный сетевой менеджер совершенно ни к чему, коли есть Systemd, говорит **Нейл Ботвик** — и учит настраивать сеть, интерфейсы и мосты.



Создавать сервер 64

Эндрю Моллетт превратил свой Raspberry Pi в веб-сервер на базе данных по технологии LEMP, применив Nginx вместо надоевшего Apache.



Управлять облаком 68

Джюлион Браун рассматривает Swift, сервис хранения объектов OpenStack, и с помощью Heat управляет инфраструктурой облака.



Рассылать письма 72

Оповестите весь мир! **Кент Ельчук** управляет почтовыми рассылками с собственного сервера (объегорив большинство спам-фильтров).



Исследовать ядро 76

Ядро Linux — такая же программа, как и все прочие. Почему бы не попробовать в ней разобраться? Тем более, с помощью **Дмитрия Кузнецова**.



Собирать схемы 80

Михаил Остапкевич и **Евгений Балдин** исследуют чудеса современной электроники; оказывается, совершать их под силу любому умельцу.

АКАДЕМИЯ КОДИНГА



И новичкам, и гуру!
Всегда полезно будет познать
нечто доселе неведомое

Создадим кластер

Джонни Бидвелл сколачивает кластер из нескольких Raspberry Pi — пусть перемалывают код на Python **с. 84**

Расширим GIMP

Написать плагин для GIMP **Джонни Бидвеллу** нипочем. Вот он и выводит ваши фото на новый уровень **с. 88**

Изучим Fortran

Джюльетта Кемп орудует элегантно-каменным топориком для программистов — он все еще работоспособен и остер **с. 92**



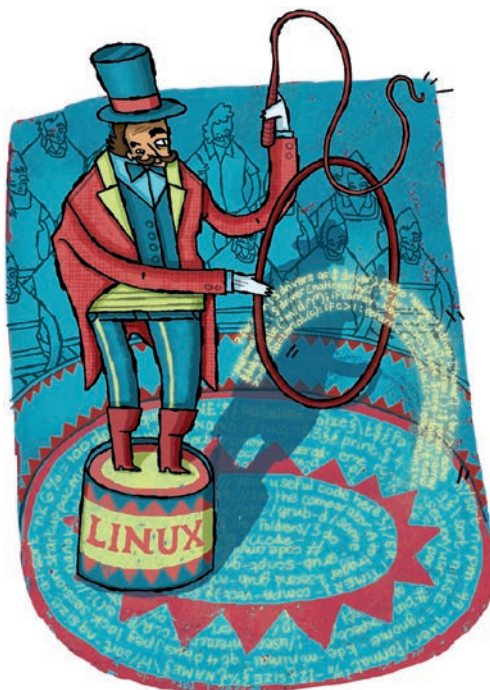
Аккаунты: Права и владение

Один у вас пользователь или сто, **Нейл Ботвик** выучит вас азам работы с учётными записями в Linux.



Наш эксперт

По жизни у **Нейла Ботвика** седьмы гораздо больше — он поседел, постоянно скрываясь от АНБ.



Linux — многопользовательская система, даже если вы работаете с компьютером в одиночку. Даже в простейшей системе есть две роли: вы и суперпользователь, он же **root**. Каждый файл или папка принадлежит какому-то пользователю и имеет параметры, называемые разрешения, которые определяют, кто обладает на него правом чтения или записи. Это не позволит другим пользователям перезаписать ваши файлы или,

при соответствующих настройках, даже их прочесть. Они также защищают системные файлы: ведь те принадлежат суперпользователю, и только он может вносить в них изменения. То же касается записи в системные каталоги — так, только суперпользователь может устанавливать туда новое программное обеспечение. А тогда как же устанавливать ПО? Ответ вы найдёте во врезке «Как стать суперпользователем?»

Предостережение при использовании данных на нескольких компьютерах, например, на внешнем жестком диске: сами вы видите свое имя пользователя как слово, скажем, johnny99, но компьютер видит и сохраняет его как число — UID, или ID пользователя. В процессе установки ваш дистрибутив создаёт суперпользователя, который всегда имеет UID 0, и обычного пользователя. Большинство дистрибутивов нумеруют первого обычного пользователя начиная с 1000, но некоторые начинают с 500. Суть в том, что на диске в качестве владельца файла хранится именно UID, и права пользователя с тем же именем могут не совпасть при переносе диска на другой компьютер.

Создание пользователя

У каждого пользователя есть своя домашняя папка. Как правило, это **home/username**, но на самом деле она может быть где угодно — у пользователя, созданного для запуска web-сервера, это будет что-то в районе **/var/www**. Помимо пользователей, в Linux есть еще и группы. По сути, группа — это набор пользователей. Например, если на вашем компьютере есть USB-сканер, зачастую нужно быть членом группы сканера, чтобы иметь возможность использовать его.

Итак, разобравшись с именами пользователей, групп, UID и домашними папками, мы можем создать пользователя:

```
sudo useradd -m -c "John Smith" -g users -G scanner,audio john
```

Как стать суперпользователем

Почти все команды управления пользователями требуют **root**-доступа. Суперпользователь, часто называемый **root**, может делать всё что угодно и где угодно, невзирая на любые разрешения. И как же скромному обычному пользователю сделать нечто требующее привилегий **root**? Есть два способа.

Традиционный заключается в использовании команды **su** (для переключения пользователя): она позволяет вам стать другим пользователем, если вы знаете его пароль. Наберите **su** в терминале (если пользователь не указан, он переключится на **root**) и введите пароль суперпользователя. Как только вы это сделаете, в данном

сеансе терминала вы — **root**, и вам всё дозволено (злорадный хохот).

Конечно, это немного рискованно, особенно если вы не переключитесь обратно сразу, как только всё закончите. Это также означает, что пароль суперпользователя, а значит, и полный доступ, попадает к любому, кому нужно выполнить команду от имени **root** — и так появилась **sudo**. Команда используется для выполнения отдельных команд с правами **root** (или другого пользователя), но не требует, чтобы вы знали пароль суперпользователя. Вместо этого, **root** должен разрешить вам запускать некоторые или все команды с его правами, так что пароль

запрашивается ваш. Обычная установка в большинстве дистрибутивов предоставляет первому созданному пользователю **sudo**-доступ ко всему, даже без входа в систему с правами администратора и передачи пароля **root**. На самом деле, в некоторых дистрибутивах вообще нет пароля для **root**, и **sudo** — единственный способ запускать команды с правами администратора.

С помощью **sudo** можно разрешить отдельным пользователям выполнять только определенные команды; плюс, она записывает все команды в системном журнале, наряду с теми, кто их выполнял, делая систему более безопасной и контролируемой.

Разные файловые системы

Много путаницы связано с разрешениями при монтировании файловой системы. Какие разрешения и права устанавливаются на точки монтирования до того, как этот процесс завершен, совершенно безразлично, поскольку их назначает корневая директория той системы, которую вы монтируете. Способ внесения этих изменений зависит от её типа. В файловых системах Linux это просто: нужно её смонтировать, а затем использовать **chmod** или **chown**, чтобы настроить всё так, как вы хотите. Эти настройки впоследствии будут применяться повсеместно, при последующем монтировании данной файловой системы.

С файловыми системами Windows, такими как FAT на USB-носителях, нужно обращаться

по-разному. В FAT нет представления о собственности файла, а система NTFS несовместима с правами Linux, поэтому драйвер файловой системы применяет разрешения по умолчанию. Они, как правило, закрепляют права на все файлы и каталоги за пользователем, смонтировавшим это устройство, иначе вы не сможете ничего туда записать. Права доступа к файлам по умолчанию закреплены за **rxr-xr-x** (или 755, если вам так больше нравится).

Если вы монтируете файловую систему вручную, используйте для установки прав опцию **uid** и **umask** для разрешений по умолчанию, вот так:

```
sudo mount /dev/sdb1 /mnt/removable -o uid=john,umask=022
```

Umask вычитается из 777, чтобы передать права, в данном случае — 755. umask, равное 0, даёт все разрешения.

```
nelz@shooty ~/Lxf/TerminalTutorials $ mount /dev/sdb1 on /media/16G type vfat (rw,nosuid,uid=1000,gid=100,umask=0022,dmask=0077,iocharset=iso8859-1,shortname=mixed,showexec,errors=remount-ro,uhelper=udisks2)
nelz@shooty ~/Lxf/TerminalTutorials $
```

➤ Команда **mount** сама по себе показывает параметры, использованные при монтировании данной файловой системы. На рисунке — параметры USB-носителя, форматированного в FAT.

Мы используем здесь **sudo**, потому что пользователей может создавать только **root**. Опция **-m** создает домашнюю папку в **/home/john**; **-c** определяет, как имя пользователя будет отображаться в комментариях — как правило, полным именем пользователя; **-g** задаёт первичную группу пользователя, а **-G** добавляет вторичные группы. И, наконец, мы определяем имя пользователя. Не все параметры обязательны: например, если вы опустите **-g**, будет использована группа по умолчанию. В одних дистрибутивах есть общая группа под названием **users** для всех несистемных пользователей, а другие создают отдельную группу для каждого. Команда **groupadd** действует аналогичным образом, равно как и оба её эквивалента — **userdel** и **groupdel**.

Добавление пароля

Пользователь создан, но он не может войти в систему, пока мы не назначим ему пароль:

```
sudo passwd john
```

Вас попросят ввести пароль дважды. Команда **passwd** может также использоваться для изменения пароля существующей учетной записи. Если запустить ее без **sudo** или имени пользователя, она изменит ваш собственный пароль, а для всех остальных пароли назначать может только **root**. Пароли рекомендуется менять регулярно, и **passwd** может вам в этом помочь:

```
sudo passwd --maxdays 60 --warndays 7 john
```

Этот пароль станет недействительным через два месяца, и Джона предупредят об истечении этого срока за неделю. И здесь, и уж тем более для суперпользователя — владельца ключей от всего королевства, действуют привычные правила

в отношении паролей: они должны быть длинными, содержать разные регистры и, желательно, цифровые символы.

Информация о пользователе хранится в **/etc/passwd** — а сам пароль, как ни странно, в **/etc/shadow** — и этот файл можно редактировать, если вы захотите что-то изменить. Однако допущенная ошибка может помешать вам войти в систему, так что для этого лучше использовать команду **vipw**. Она загружает копию **/etc/passwd** в выбранный вами редактор (как указано в **\$EDITOR**) и проверяет его правильность при сохранении, прежде чем заменить существующий файл. Формат **/etc/passwd** полностью объясняется с помощью

```
man 5 passwd
```

Передача прав

Чтобы изменить владельца файла, вам понадобится **chown**:

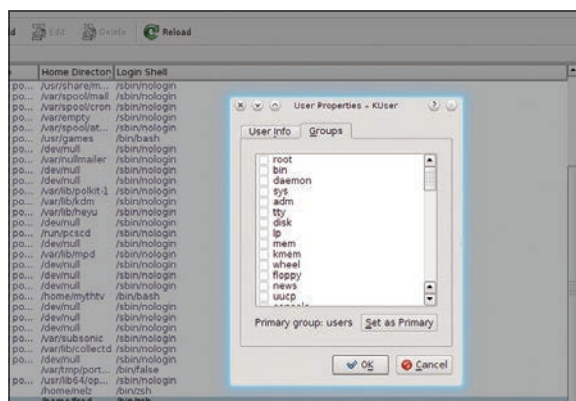
```
chown john somefile
```

```
chown john:users someotherfile
```

```
chown john: someotherfile
```

```
chown -R john: somedir
```

Первая команда делает Джона владельцем одного файла. Вторая команда также изменяет группу. Если вы не указываете группу после двоеточия, как в третьем примере, группа меняется на группу пользователя по умолчанию. Применив опцию **-R** к папке, вы замените все находящиеся в ней файлы и подкаталоги. Кроме того, вы можете изменить только группу, с **chgrp**. Эти команды нельзя применять без привилегий суперпользователя. Изменение прав доступа к файлам осуществляется с **chmod**, о чём уже говорилось ранее в этой серии учебников. LXF



➤ Большинство рабочих столов предлагает графические альтернативы для управления пользователями; здесь — **Kuser** от KDE.

```
I /etc/sudoers.tmp (Modified Row 54 Col 1 1:59 Ctrl-K H for help)
# *** REMEMBER *****
# * GIVING SUDO ACCESS TO USERS ALLOWS THEM TO RUN THE SPECIFIED *
# * COMMANDS WITH ELEVATED PRIVILEGES. *
# *
# * NEVER PERMIT UNTRUSTED USERS TO ACCESS SUDO. *
# *****
# User privilege specification
root ALL=(ALL) ALL
# Uncomment to allow people in group wheel to run all commands
wheel ALL=(ALL) ALL
admin ALL=(ALL) ALL
# Same thing without a password
#wheel ALL=(ALL) NOPASSWD: ALL
# Users in group www are allowed to edit httpd.conf and ftpd.conf
```

➤ **sudo** раздает всем пользователям права управления системой — полные, ограниченные или вовсе никаких; здесь — типовое определение администратора (ему можно всё).

Лучше: Свой сервер фото

Запустите личный легковесный сетевой Flickr и делитесь своими картинками.



Наш эксперт

Маянк Шарма был редактором *Linux.com*, и в *LXF186* рассказал, как превратить Raspberry Pi в музыкальный сервер.



Есть множество онлайн-сервисов, через которые можно поделиться своими фотографиями без малейших затрат. Однако такие сайты в первую очередь предназначены для того, чтобы делиться фотографиями с вашей социальной сетью онлайн. И хотя у некоторых из них тонко настроены опции конфиденциальности, все же это не идеальное место, чтобы выставлять

свои более личные фотографии для друзей и родных. Гораздо лучше будет хранить личные фотографии и делиться ими, не теряя над ними контроля — на собственном хостинге. Это потребует дополнительных усилий, но зато вы сами будете отвечать за свои фотографии, а когда хостинг будет полностью настроен, у вас будет практически тот же уровень удобства, которого можно ожидать от ведущих сервисов хостинга фото.

Чтобы развернуть ПО web-альбома, нужен web-сервер, а с этой задачей вполне достойно справится даже Raspberry Pi. Очень популярен web-сервер *Apache*, но мы здесь запустим легковесный web-сервер *Nginx* [на стр. 64 есть очень подробное руководство], чтобы наилучшим образом использовать ограниченные ресурсы таких систем, как Pi.

Чтобы разместить наши цифровые фотографии, управлять и делиться ими, воспользуемся web-приложением *Lychee*. По сравнению с другими программами, предлагающими похожие функции, *Lychee* легковесно, просто в настройке и имеет интуитивно понятный и приятный пользовательский интерфейс.

Начните с запуска и обновления своего Raspberry Pi через **sudo apt-get update**. Вы можете настроить *Lychee* с удаленного компьютера, зайдя на Pi через SSH с удаленного компьютера в сети.

Для данного проекта будем предполагать, что IP-адрес Raspberry Pi — **192.168.2.100**. Вы можете зайти на него с удаленного компьютера по

```
sudo ssh pi@192.168.2.100
```

Введите команду, и для входа у вас спросят пароль пользователя pi.

Когда список пакетов обновится, установите язык скриптов PHP и требуемые библиотеки:

Расширяем Lychee

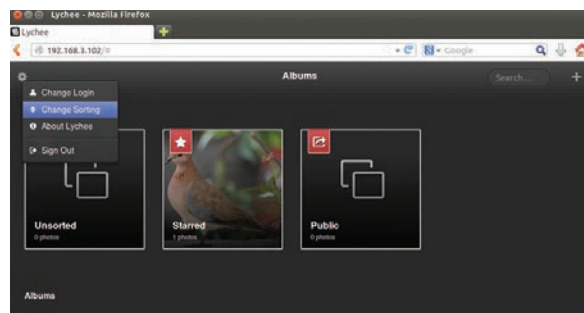
Вы можете расширить свой *Lychee* так, чтобы он автоматически синхронизировался с сервисом облачного хранения, например, Google Drive или Dropbox.

Поскольку ни один из этих сервисов не имеет официального клиента для Raspberry Pi, придется использовать индивидуальные скрипты. Загляните на <http://robotic-controls.com/learn/raspberry-pi-syncing-files-raspberry-pi>, и вы найдете различные способы синхронизации директории на Raspberry Pi с онлайн-хранилищем.

Если вы хотите синхронизировать фотографии с учетной записью Google Drive, вам пригодится скрипт *Grive*. На странице

есть подробные инструкции по компиляции этого скрипта из исходника, поскольку его нет в официальном Raspbian. Установив скрипт, запустите его из папки, где хранятся ваши изображения, и он выдаст вам URL, где предоставляется код аутентификации, нужный вам перед синхронизацией файлов.

Аналогично, вы можете использовать скрипт *dropfuse*, чтобы поделиться своими изображениями с определенной папкой Dropbox. Этот скрипт не дает распределенного доступа ко всему вашему хранилищу Dropbox, и может соединяться только для передачи файлов в определенную папку.



» Опций настройки в *Lychee* голый минимум, но есть расширение командной строки, которое сканирует и добавляет любую директорию с фотографиями как альбом.

```
sudo apt-get install php5-fpm php5-gd libgd2-xpm libjpeg0
libxpm4
```

Затем установите программы web-сервера:

```
sudo apt-get install nginx
```

И, наконец, установите сервер базы данных *MySQL* и привяжите его к PHP:

```
sudo apt-get install mysql-server php5-mysql
```

Во время установки вам предложат задать пароль пользователя *root* для *MySQL*. Этот пароль понадобится вам позднее, чтобы создать базу данных для *Lychee*.

Теперь отредактируем файл настройки PHP, чтобы изменить значение некоторых параметров с их консервативных значений по умолчанию и пользоваться *Lychee* без ограничений. Откройте файл настройки в текстовом редакторе *nano*:

```
sudo nano /etc/php5/fpm/php.ini
```

Затем с помощью *Ctrl+W* найдите нижеследующие переменные и задайте им такие значения взамен значений по умолчанию:

```
max_execution_time = 200
```

```
post_max_size = 200M
```

```
upload_max_size = 200M
```

```
upload_max_filesize = 20M
```

```
max_file_uploads = 100
```

Для тех переменных, которых в вашем файле не существует, добавляйте новый пункт.

Покончив с этим, прокрутите список до конца файла и добавьте следующие строки, чтобы активировать требуемые расширения:

```
extension = php_mbstring.dll
```

```
extension = php_exif.dll
```

```
extension = php_gd2.dll
```

Сохраните и выйдите с помощью *Ctrl+X*, и перезапустите сервис PHP:

```
sudo service php5-fpm restart
```

Настроив компоненты инфраструктуры, создайте директорию для размещения файлов для *Lychee* с помощью **sudo mkdir /var/www**. Затем перейдите в эту директорию и установите программу *Git* через **sudo apt-get install git**.

Используйте программу, чтобы найти самую свежую версию *Lychee*:

```
git clone https://github.com/electerious/Lychee.git.
```

Файлы скачаются в директорию **/var/www/Lychee**.

Когда файлы будут скачаны, передайте владение директорией пользователю web-сервера:

```
sudo chown -R www-data:www-data /var/www/Lychee
```

Разрешите также запись по всему миру для закачанных материалов, а также папки PHP и файлов внутри с помощью **chmod -R 777 uploads/ php/**.

Когда все программы будут на месте, нужно сообщить *Nginx* об их существовании. Для этого отредактируйте файл настройки:

```
sudo nano /etc/nginx/sites-available/default
```

и добавьте следующее, чтобы указать на вашу установку *Lychee*:

```
Server {
    root /var/www/Lychee;
    index index.php index.html index.htm;
    location ~ \.php$ {
        fastcgi_pass unix:/var/run/PHP5-fpm.sock;
        fastcgi_index index.php;
        include fastcgi_params;
    }
}
```

Перезапустите web-сервер с помощью **sudo service nginx restart**, чтобы изменения настройки вступили в силу. Теперь откройте браузер и укажите ему на установленный *Lychee* на вашем Raspberry Pi на **192.168.2.100** или по другому адресу.

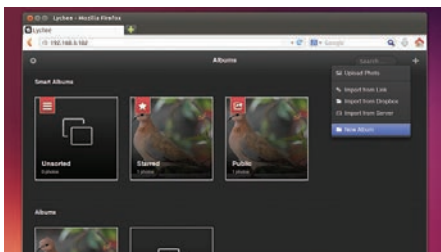
Поскольку это первичная установка, *Lychee* наскоро проведет вас по процессу настройки. На первом экране вам сообщает подробности соединения и аутентификации для сервера базы данных *MySQL*. В качестве хоста используйте **localhost**, а в качестве имени пользователя — **root**. В поле пароля введите пароль, заданный при установке сервера *MySQL*.

Вы также можете предложить *Lychee* использовать существующую базу данных, хотя лучше будет предоставить создание базы данных самому *Lychee*. Нажмите **Connect**, чтобы перейти ко второй стадии установки. Теперь вам предложат создать имя пользователя и пароль для доступа и администрирования этой установки *Lychee*. Затем нажатие **Create Login** завершит процесс.

Вот и все. *Lychee* соединится с сервером *MySQL* и создаст базу данных для себя, а также регистрационную информацию для вас, которую вы сможете использовать, чтобы загружать свои фотографии и управлять ими. После этого *Lychee* автоматически переведет вас в основной интерфейс. Отсюда вы можете использовать программу, чтобы создавать альбомы, загружать и делиться фотографиями и администрировать свою установку *Lychee*.

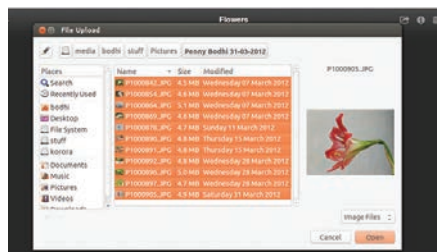
»

Переносите фотографии в Lychee



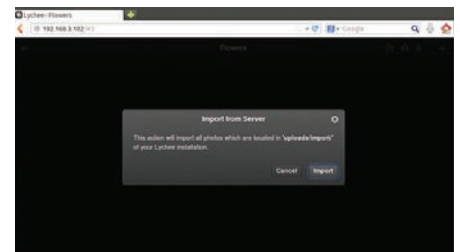
1 Создайте альбом

Чтобы загрузить свои фотографии, щелкните по значку '+' в правом верхнем углу web-интерфейса *Lychee*. Если вы пока что не рассортировали свои фотографии по категориям, можете использовать опцию **Upload Photo**, чтобы импортировать фотографии в *Lychee*, а затем распределить их по альбомам. Все фотографии, не загруженные в определенный альбом, размещаются в «умном» альбоме 'Unsorted'.



2 Загружайте с диска

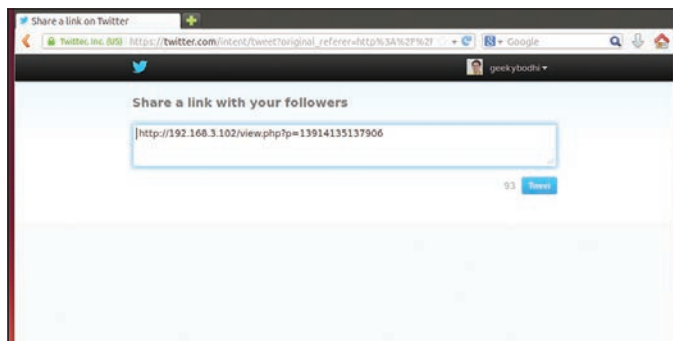
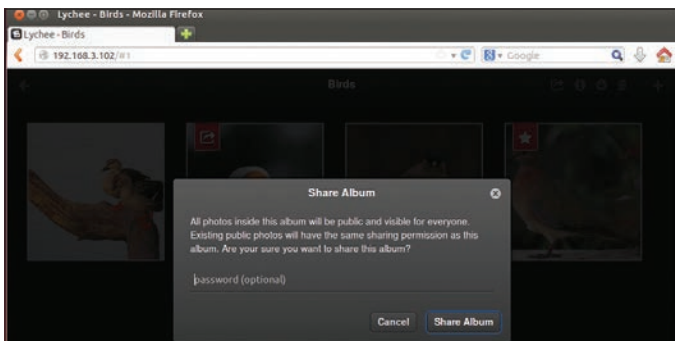
Если вы хотите создать альбом, можете также использовать опцию **New Album**. Опция **Upload Photo** запустит менеджер файлов — он используется для навигации по фотографиям, которые вы хотите загрузить. Вы можете выбрать отдельную фотографию или выбрать сразу несколько при помощи клавиш **Shift** и **Ctrl**. Затем *Lychee* загрузит их на сервер и обработает их.



3 Импорт

Помимо добавления фотографий с диска, *Lychee* умеет импортировать их из других источников. У значка '+' есть опции прямого импорта фотографий с URL и **Dropbox**. По своему усмотрению вы также можете применить любой клиент FTP для передачи изображений в директорию *Lychee* для закачек/импорта и затем использовать опцию **Import from Server** для импорта любых изображений, размещенных здесь.

Делитесь фотографиями с друзьями и родными

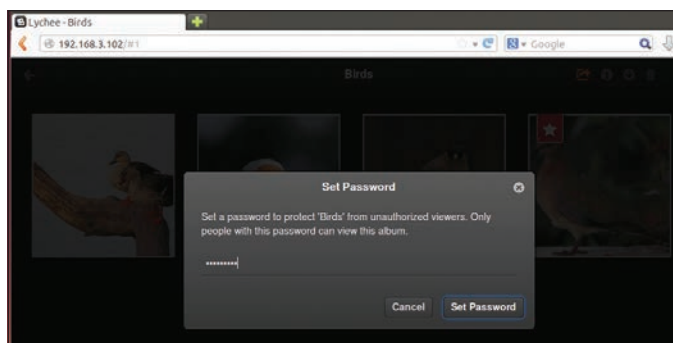
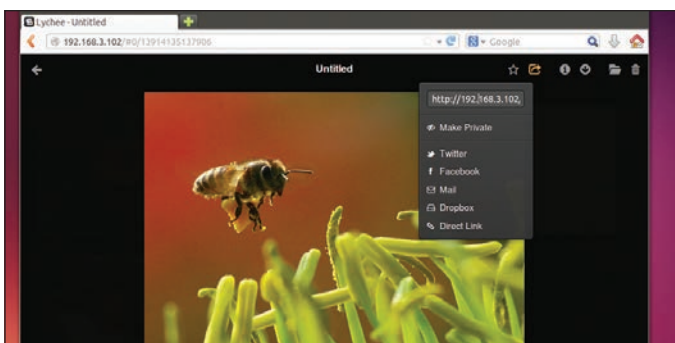


1 Поделитесь альбомом

Просматривая альбом, можете щелкнуть по значку 'share' в верхнем правом углу интерфейса. Он выдаст вам довольно длинный URL, который вы сможете передать другим. Но помните, что этот URL будет доступен только для пользователей вашей сети. Если вы хотите разрешить и другим пользователям просматривать этот URL, вам придется проделать дыры в брандмауэре вашего роутера, открыв доступ.

2 Делитесь легко

Для упрощения процесса передачи URL можете поделиться им в Twitter или в Facebook прямо из *Lychee*. Значок share предлагает опции размещения URL напрямую как твит или обновление в Facebook. Более того, если вы используете опцию Mail, *Lychee* запустит ваш почтовый клиент по умолчанию и создаст для вас новое сообщение, содержащее URL альбома, которым вы хотите поделиться.

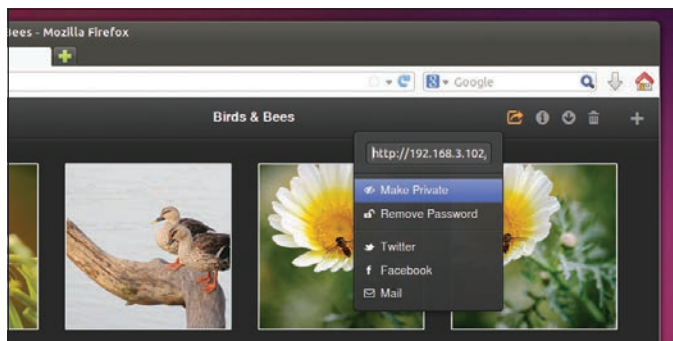
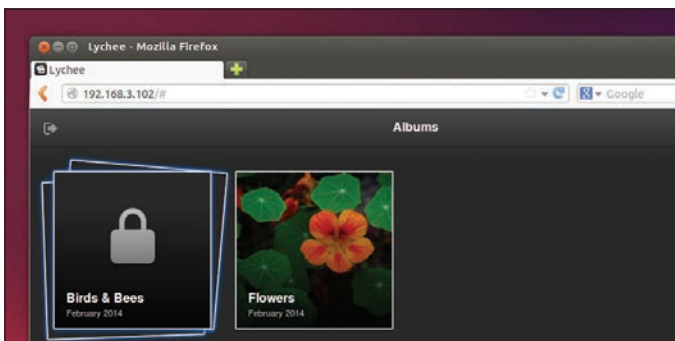


3 Делитесь фотографиями

Так же, как альбомами, можно поделиться отдельными фотографиями, выбрав значок share при просмотре отдельных изображений. И опять-таки, как и с альбомами, у вас есть опция поделиться URL'ом фотографии в Twitter, Facebook или по электронной почте. Кроме того, есть опция загрузки изображения в Dropbox, а также опция отображения прямого URL фотографии в новой вкладке.

4 Защита паролем

Для укрепления безопасности, когда вы делитесь альбомом, *Lychee* попросит вас ограничить доступ к нему, защитив его паролем. Это чисто опционально, и это можно пропустить. Но, решив ограничить доступ паролем, вы найдете опцию создания пароля в значке share. Помните, что вы можете защитить паролем только альбомы, а не отдельные фотографии, которыми вы делитесь.



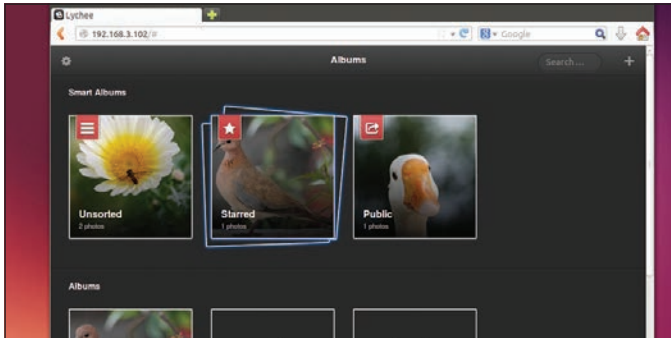
5 Общественный альбом

Когда вы делитесь фотографией, *Lychee* ставит на ее эскиз значок, указывающий на доступность для публичного просмотра. Кроме того, для более простого доступа программа группирует все альбомы с разрешенным доступом и фотографии в альбоме 'Public'. Публичные альбомы видны всем пользователям в вашей сети на загрузочной странице вашего сервера *Lychee* — в нашем случае, 192.168.2.100.

6 Перестать делиться

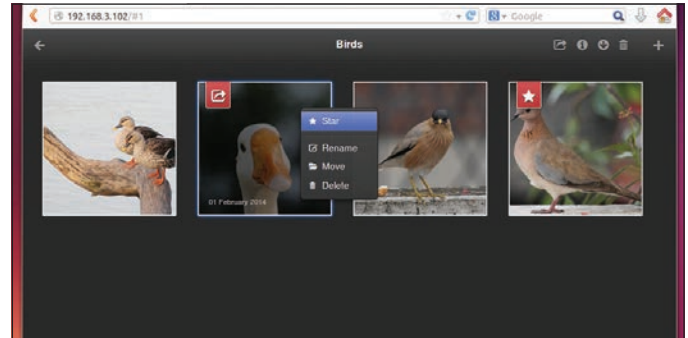
Опция Make Private позволяет отозвать разрешение для альбома, которым вы поделились, или отдельного фото. Когда вы прекращаете делиться альбомом или фото, *Lychee* автоматически удаляет его из альбома 'Public' и удаляет его значок публичности. Учтите, что в публичном альбоме нельзя сделать отдельные фото частными: функцию видимости придется изменить для всего альбома.

Сортируйте и редактируйте свои альбомы



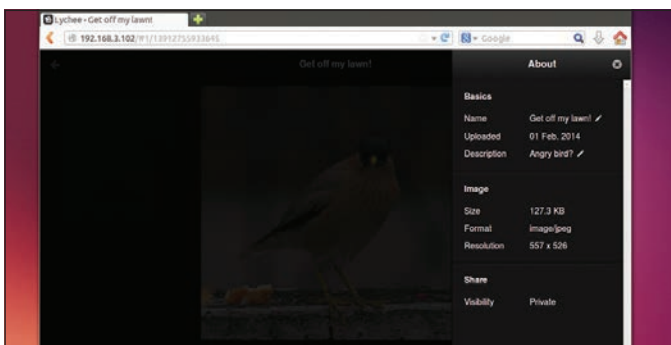
1 Умные альбомы

Для более простого управления в *Lychee* есть ряд виртуальных директорий, именуемых «умными [smart] альбомами» — они размещают фотографии, не обязательно находящиеся в одной директории. Одна такая директория называется 'Unsorted' и размещает все фото, которые не сгруппированы ни в один альбом. Альбом 'Starred' показывает изображения, которых вы пометили оценкой, а в директории 'Public' находятся изображения, которыми вы поделились.



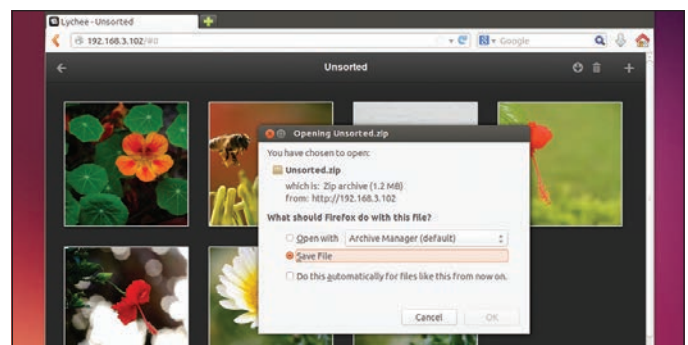
2 Подготовьте фотографии

Загрузив изображение в *Lychee*, вы получаете несколько опций для их достойного размещения. В правом верхнем углу основного экрана интерфейса имеется окно Search, помогающее в поиске изображения. Просматривая галерею, вы можете щелкнуть правой кнопкой по фото и получить доступ к функциям, которые позволяют переименовать его или переместить в другой альбом. Отсюда также можно удалить фото с сервера или выделить его.



3 Показывайте и правьте данные EXIF

У большинства изображений есть ассоциированные с ними метаданные, называемые данными EXIF. Вместе с изображениями, *Lychee* также импортирует и эти метаданные. Чтобы просмотреть EXIF-данные изображения, откройте его и щелкните по значку 'i' в верхнем правом углу. Помимо отображения данных EXIF, вы также можете щелкнуть по значку с карандашом, чтобы отредактировать определенную часть информации — например, имя и описание.



4 Передача фотографий

С помощью щелчка по зеленой стрелке, указывающей вниз, вы можете скачать отдельные фотографии в полном разрешении, а также целые альбомы в сжатом архиве ZIP. Чтобы упростить передачу фотографий в *Lychee*, вы можете также воспользоваться клиентом FTP и загрузить их в директорию **upload/import** в *Lychee*. В веб-интерфейсе предусмотрена опция импорта изображений именно из этой директории. **LXF**

Используйте свой Pi в качестве фоторамки

Установив Raspbian и обновив его репозитории с помощью `sudo apt-get update`, установите *Quick Image Viewer* командой

```
sudo apt-get install qiv
```

Теперь создайте директорию **images/** в домашней директории (**/home/pi**) и перенесите все изображения сюда. Затем откройте текстовый редактор и создайте файл под названием **photoframe**, со строкой

```
qiv --fullscreen --slide --random --delay=10 /home/pi/images/*
```

Она запустит все изображения в директории в случайном порядке и будет отображать каждое в полноэкранном режиме в течение 10 секунд. Сохраните файл, выйдите из текстового редактора и сделайте скрипт исполняемым с помощью `chmod +x photoframe`. Затем найдите файл под названием **.xinitrc** в вашей домашней директории (**/home/pi**),

а если такого файла не существует, создайте его и добавьте в него следующую строку:

```
/home/pi/photoframe
```

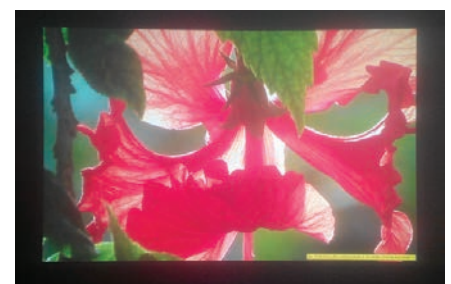
Запустите X-сервер командой **startx**, и он автоматически запустит слайд-шоу.

Если у вас много изображений, можете использовать USB-диск, но проверьте, чтобы он был монтирован в том же месте, чтобы скрипт мог найти изображения.

Подключите диск и найдите его местоположение, просканировав несколько последних строк вывода **dmesg**. Найдите уникальный UUID этого диска с помощью `ls -l /dev/disk/by-uuid/` и создайте точку монтирования для диска с помощью `sudo mkdir /media/usb-images`.

И, наконец, добавьте следующее в **/etc/fstab**, чтобы автоматически подмонтировать этот диск: `UUID=E352-47DC /media/usb-images vfat default 0 2`.

Таким образом вы гарантируете, что устройство USB всегда будет монтироваться на **/media/usb-images**. Теперь можете указать это местоположение в скрипте фоторамки.



➤ Наслаждайтесь слайд-шоу из своих фото на Raspberry Pi и настройте его по своему вкусу.

Systemd. СВЯЗИ НАДОЖИВАЮТСЯ

Нейл Ботвик считает, что если вы — фанат *Systemd*, вам, вероятно, больше незачем использовать отдельный менеджер сети.



Наш эксперт

У **Нейла Ботвика** по компьютеру в каждой комнате, но по соображениям безопасности он нипочем не расскажет, где находится центральный сервер.

Мы рассматривали *Systemd* в **LXF184**. Там мы упомянули о новом инструменте сетевого управления, *Systemd-networkd*, который только что появился. Мы не вдавались в подробности, потому что он был совсем новым и ни один из дистрибутивов не использовал эту версию. С тех пор вышло еще пять обновлений *Systemd*. Сетевой менеджер быстро взрослеет и доступен во многих дистрибутивах, так что пора задуматься о его использовании.

Прежде чем двигаться дальше, проясним применение некоторых неоднозначных терминов. Говоря о менеджере сети, мы имеем в виду любую программу, используемую для управления вашим сетевым соединением. А говоря о *NetworkManager*, мы имеем в виду совершенно конкретную программу управления сетью от Gnome, в противоположность таким программам, как *ConnMan*, *Wicd* или иным инструментам настройки сети, предлагаемыми другими дистрибутивами.

Systemd-networkd — это простой и легковесный менеджер сети. Он не предназначен для замены собой функциональности *NetworkManager*, который работает с проводными и беспроводными соединениями, соединениями 3G, PPPoE и VPN. Его главная цель — удаленно управляемые системы с простыми требованиями к сети, которым не нужны все зависимости всестороннего графического менеджера. И не такой уж он простак, поскольку умеет работать с несколькими интерфейсами, сетями и режимами моста.

Многие системы работают с одним сетевым интерфейсом, настроенным или статически, или с помощью DHCP, и их будет идеально отдать на откуп *Systemd* — зачем устанавливать лишний набор программ и зависимостей, если у вас уже есть системная программа управления, которая делает эту работу?

Первые шаги

Сети в *Systemd* определяются файлами unit, так же, как *Systemd* определяет все остальное. Эти файлы unit имеют расширение **.network** и хранятся в **/etc/systemd/network**. Когда обнаруживается

сетевой интерфейс, соответствующий одному из ваших файлов **.network** — или при запуске *Systemd-networkd*, или в любое время после — применяется настройка из файла. Если *networkd* останавливается или перезапускается, большинство соединений остаются нетронутыми, что в некоторых ситуациях бывает удобно.

Самая простая конфигурация — одиночный проводной интерфейс, соединенный с роутером, который получает свой адрес от DHCP-сервера роутера. Давайте сделаем это в первую очередь, создав такой файл **/etc/systemd/network/wired.network**:

```
[Match]
Name=eth0
[Network]
Description=Wired network
DHCP=yes
```

Здесь содержится два раздела. Раздел **Match** указывает устройство, к которому будет применена эта настройка, в данном случае — одиночный интерфейс Ethernet eth0. Можно использовать другие свойства устройства, такие, как **MACaddress**, **Driver** или **Type**. Им соответствуют свойства **udev**. Можно применять маски: так, **Name=en*** соответствовал бы любой карте Ethernet, использующей новый протокол присваивания имен. Если у вас в разделе **Match** много пунктов, все они должны соответствовать применяемому устройству. Если интерфейсу соответствует более одного файла **.network**, берется только найденный первым, поэтому было бы разумно нумеровать файлы, чтобы контролировать порядок их считывания.

Раздел **Network** работает с конфигурацией любого подходящего устройства. Здесь у нас две настройки: **Description** — опциональный пункт, это скорее комментарий, чем настройка. **DHCP=yes** дает команду *Systemd-networkd* предложить серверу DHCP настроить интерфейс, что вам обычно и требуется.

Единственный другой шаг — убедиться, что сервис *networkd* запускается, с помощью

```
systemctl start systemd-networkd
systemctl enable systemd-networkd
```

Убедившись, что все работает так, как вам надо, заставьте сервис запускаться при загрузке, через команду

Не забудьте отключить любой другой инструмент сетевого управления, которым вы, возможно, пользуетесь.

Настройка статической сети немногим сложнее. Нужно указать адрес, шлюз и DNS-сервер(ы) для использования:

```
[Match]
Name=eth0
[Network]
Address=192.168.2.5/24
Gateway=192.168.2.42
DNS=192.168.2.42
```

» Используя файлы настройки в **/etc/systemd/system**, вы можете обеспечить, чтобы у *Networkd* все нужные ему файлы и зависимости были на месте.

```
[root@shooty systemd/system 0]% cat systemd-networkd.service
[Service]
ExecStartPost=/bin/ln -sf /run/systemd/resolve/resolv.conf /etc/resolv.conf
ExecStopPost=/bin/rm -f /etc/resolv.conf
ExecStopPost=/usr/bin/touch /etc/resolv.conf
ExecStopPost=-/bin/cp -pf /etc/resolv.conf.bak /etc/resolv.conf
[root@shooty systemd/system 0]%
[root@shooty systemd/system 0]%
[root@shooty systemd/system 0]% ls -l --color systemd-networkd.service
total 4
lrwxrwxrwx 1 root root 49 Jun 23 09:07 wpa_supplicant.service
64/systemd/system/wpa_supplicant@.service
[root@shooty systemd/system 0]%
```


Для более сложных настроек есть отдельные разделы **Address** и **Route**, но для подобного рода простой сети вы можете разместить все в **Network**. Man-страница `systemd.network` все объясняет более подробно и поможет вам выбраться, если вам понадобится чуть больше содействия.

Переходим к беспроводному

Хотя `networkd` не предназначен для работы на ноутбуках, все же это хороший выбор для целого ряда ситуаций. Если вы постоянно скачете из одной сети в другую, для вас, вероятно, лучшей опцией станет более всесторонний менеджер сети. Но если вы соединяетесь только с ограниченным количеством сетей, `Systemd-networkd` облегчит вам жизнь. Процесс соединения с открытой сетью точно такой же, как при проводном соединении: нужно только изменить раздел **Match**, чтобы он соответствовал вашему беспроводному интерфейсу. Это общий сценарий для беспроводных точек доступа: вы соединяетесь без аутентификации и затем заходите на web-страницу.

Если вы хотите соединиться с сетью, использующей шифрование WPA, нужно воспользоваться `wpa_supplicant`, который `Systemd` запустит для вас. Добавьте сеть в файл настройки `wpa_supplicant` командой

```
wpa_passphrase «ESSID» «passphrase» | sudo tee -a /etc/wpa_supplicant/wpa_supplicant.conf
```

Повторите это для каждой добавляемой сети. Если вы не знаете ESSID сети, можно получить его через `iwlist`:

```
sudo iwlist wlan0 scan | less
```

Создайте файл сети для беспроводного интерфейса точно так же, как и для проводного интерфейса с DHCP. Чтобы `wpa_supplicant` работал на `wlan0`, запустим

```
sudo systemctl start wpa_supplicant@wlan0.service
```

Он будет искать настройку в файле `/etc/wpa_supplicant/wpa_supplicant-wlan0.conf`. Вы можете заменить соответствующие определения сети из `wpa_passphrase` в этом файле или поместить все в `/etc/wpa_supplicant/wpa_supplicant.conf` и создать символическую ссылку [symlink] в `/etc/wpa_supplicant/wpa_supplicant-wlan0.conf`. Вы могли бы заставить сервис `wpa_supplicant` запускаться при загрузке, через

```
sudo systemctl start wpa_supplicant@wlan0.service
```

но поскольку `wpa_supplicant` нужен только тогда, когда работает `Systemd-networkd`, лучше обеспечить его запуск каждый раз при запуске `Systemd-networkd`, такими командами:

```
sudo mkdir /etc/systemd/system/systemd-networkd.service.wants
sudo ln -s /usr/lib/systemd/system/wpa_supplicant@.service
/etc/systemd/system/systemd-networkd.service.wants/
wpa_supplicant@wlan0.service
```

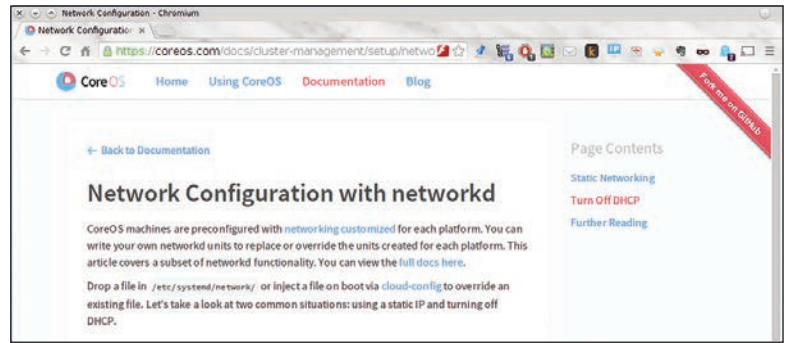
Серверы имен

`Systemd` не обновляет традиционный файл `/etc/resolv.conf`: вместо этого он хранит информацию сервера имен в `/run/systemd/resolve/resolv.conf`. Чтобы она стала доступной для всей остальной системы, `/etc/resolv.conf` следует сделать символической ссылкой на этот файл. Если вы используете для работы с сетью только `Systemd`, вы можете просто создать эту ссылку и забыть о ней:

```
sudo ln -sf /run/systemd/resolve/resolv.conf /etc/resolv.conf
```

Если вы используете любую другую форму управления сетью, вам нужна эта ссылка только когда работает `Systemd-networkd`. Файлы `unit` `Systemd` позволяют указать команды, выполняемые при запуске и остановке сервиса, поэтому создайте директорию `/etc/systemd/system/systemd-networkd.service.d` и файл, оканчивающийся на `.conf`, содержащий

```
[Service]
ExecStartPost=/bin/ln -sf /run/systemd/resolve/resolv.conf /etc/resolv.conf
```



► **CoreOS** — серверный дистрибутив, который подталкивает разработку `Networkd`. Вы найдете больше документации по нему на его сайте.

```
ExecStopPost=/bin/rm -f /etc/resolv.conf
```

```
ExecStopPost=/usr/bin/touch /etc/resolv.conf
```

```
ExecStopPost=-/bin/cp -pf /etc/resolv.conf.bak /etc/resolv.conf
```

Первая строка создает символическую ссылку при запуске. Остальные запускаются, когда сервис останавливается, и они удаляют ссылку, создают пустой файл и затем замещают его файлом резервной копии, если он существует (дефис, предшествующий финальной команде, велит ей игнорировать любую возникшую ошибку). Это создаст вам установки по умолчанию, когда `Systemd-networkd` не работает.

Режим моста

С ростом использования виртуализации, режим моста становится все более и более распространенным, и `networkd` успешно с ним работает. Другие сетевые файлы, которые мы рассмотрели, работали с существующими устройствами. Однако с режимом моста нам надо сначала создать виртуальное сетевое устройство, которое использует файл `.netdev`, в том же месте, что и файлы `.network`:

```
[NetDev]
Name=br0
Kind=bridge
```

При запуске `networkd` создается устройство `br0`. Теперь создайте мост с помощью файла `.network`:

```
[Match]
Name=br0
[Network]
DHCP=yes
```

Вышеупомянутая команда использует DHCP, но может использовать и статическую конфигурацию. И, наконец, добавьте к мосту одно или более устройств с помощью отдельного файла `.network` для каждого:

```
[Match]
Name=eth0
[Network]
Bridge=br0
```

Учтите: каждый файл описывает конфигурацию только одного устройства. **LXF**

Мировое господство

Есть такая теория, что `Systemd` норовит завоевать весь мир. Нет, это неправда: мировое господство — амбиция Линуса Торвальдса. `Systemd` обретает все новые возможности и умения, но `Networkd` затевался не командой `Systemd`. За ним стояла CoreOS, которой был нужен легковесный

инструмент настройки сети для серверов, и там решили, что `Systemd` — то, что надо. Не желая использовать `Networkd`, вы ничего не потеряете; ну разве что самую малость пространства на диске, поскольку никакой код не загрузится, пока вы не запустите демон `Networkd`.

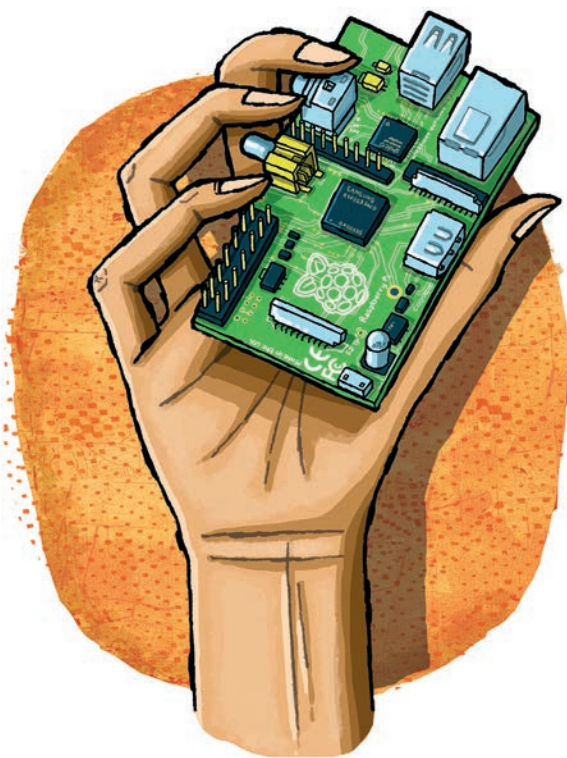
Nginx: Убийца Apache

Эндрю Моллетт использует свой крошечный Raspberry Pi, чтобы показать, как легко создать web-сайт с базой данных по технологии LEMP.



Наш эксперт

Эндрю Моллетт — преподаватель по Linux с более чем 700 видеороликами на YouTube (<http://bit.ly/UrBPeng>). Его курсы также можно найти на www.pluralsight.com.



В Raspberry Pi удивительно то, как много возможностей уместается на столь маленькой плате. Pi может принести вам массу удовольствия, так как позволяет развлекаться и одновременно изучать что-нибудь интересное или полезное. На нашем уроке мы посмотрим, как установить на Raspberry Pi web-сервер. Вряд ли найдется много организаций, которые будут разворачивать свою серверную ферму с доступом извне только на малютках Raspberry Pi; но технологии и программы, которыми будут пользоваться они и вы, одни и те же.

Наверное, вы слышали или читали о технологии LAMP (Linux, Apache, MySQL и PHP),

поэтому мы немного уйдем в сторону и воспользуемся LEMP, где на место Apache пришел Nginx. Да, 'E' действительно означает Nginx, web-серверную технологию, название которой производится как 'Engine-x' и удобно позволяет нам добавить в аббревиатуру гласную букву; кроме того, LNMP выглядит несколько

глуповато. Объединив Linux с этими ценными ингредиентами на Raspberry Pi, мы сможем создавать динамический web-контент, который будет загружать и сохранять содержимое из серверной базы данных. На нашем уроке мы сосредоточимся на установке и разворачивании сайта, который проиллюстрирует идею. Задание базовых настроек web-сервера позволит нам развернуть простую страницу на PHP, содержимое которой будет загружаться из базы данных. Проясним, что мы будем использовать модель Raspberry Pi (Model B) 512 МБ с Debian Wheezy 7.5 на SD-карте объемом 4 ГБ; но наши указания подойдут ко всем моделям Pi.

Установка Nginx

Для начала нам нужно установить web-сервер Nginx в свою тестовую систему. Nginx — новая web-серверная технология, но по последним опросам NetCraft (www.netcraft.com) мы видим, что Интернет немного охладил к Apache, и доля Nginx с его появления в 2004 году регулярно повышается. При этом в мае 2014 года доля Apache все еще составляла 37%, Microsoft — 33% и Nginx — 14%. Чтобы начать установку, поработаем в командной строке:

```
sudo apt-get update
sudo apt-get install nginx
```

Web-сервер мы установили, но все еще не совсем готово. Нужно выполнить небольшие предварительные настройки — не бойтесь, это нетрудно. Конфигурационные файлы Nginx живут в каталоге /etc/nginx. Поэтому давайте получим привилегии root и перейдем в этот каталог:

```
sudo -i ##дает нам оболочку root, если у нас есть такие права
cd /etc/nginx
ls --color-auto
```

По выводу этой команды мы увидим, что в /etc/nginx есть два каталога. Это sites-enabled и sites-available. Все сайты, размещенные на web-сервере Nginx, будут находиться в каталоге sites-available. Каталог sites-enabled будет содержать символические ссылки на сайты из первого каталога, которые мы хотим использовать на сервере. Это типично для систем на основе Debian, и в Apache для Debian все точно так же.

Начнем с отключения сайта по умолчанию и создадим собственный сайт. Создать сайт, или виртуальный хост, в Nginx очень просто, и мы можем быть уверены в его конфигурации, так как создали его сами. С помощью следующей команды, запущенной с привилегиями root, можно удалить из каталога /etc/nginx символическую ссылку по умолчанию.

«Создать сайт, или виртуальный хост, в Nginx очень просто.»

Безопасность MySQL

Настройки безопасности по умолчанию во многих системах обычно не слишком надежны — мы наблюдали это в системах на базе Debian, например, в Raspbian, когда нам предлагали задать пароль root для MySQL. Этот пароль надо задавать обязательно, и он должен отличаться от пароля обычного пользователя root в Linux. Это — важные первые шаги повышения безопасности MySQL. Также

в нашей системе следует запустить `mysql_secure_installation`. Здесь мы этого для краткости не делаем, но учтите: осторожность лишней не бывает.

Запустив программу, вы попадете в простую командную строку. Сначала у вас спросят пароль MySQL и предложат его изменить. Затем вам предложат удалить анонимных пользователей. Если они вам не нужны, удалите их. После этого вам будет

задан вопрос, хотите ли вы отключить удаленный доступ к MySQL от имени root — лучше ответить «да». Потом вам порекомендуют удалить тестовую базу данных. Это обычно для MySQL не только для Raspberry Pi, так что не стоит об этом беспокоиться. В конце работы скрипта перезагрузится таблица прав доступа, и на этом работа утилиты будет завершена.

Затем мы создаем новый корневой каталог документа для хранения нового контента:

```
unlink sites-enabled/default
mkdir /var/www
```

Изготовим новый сайт в текстовом редакторе, создав файл `/etc/nginx/sites-available/main`:

```
server {
    listen 80;
    root /var/www;
    index index.html;
}
```

Чтобы сайт заработал, в каталоге `available` нужно, кроме самого сайта, самостоятельно создать символическую ссылку на сайт. Главная конфигурация включает все файлы из каталога `sites-enabled`. Зайдя с правами root в каталог `/etc/nginx/sites-enabled`, мы создадим символическую ссылку на главный конфигурационный файл сайта. Мы также можем создать индексную страницу сайта, пусть и со скромным содержанием. Файл `index.html` был указан в качестве стартовой страницы в настройках сайта в директиве `'index'`. Это страница по умолчанию, которая будет открываться, если конкретный URL не указан:

```
ln -s ../sites-available/main
echo "Welcome to nginx" > /var/www/index.html
```

Проверка и запуск

Теперь мы готовы к тому, чтобы запустить свои движки — по крайней мере, *Nginx*. Нужно делать это внимательно и проверять конфигурацию на отсутствие ошибок. Это всегда хорошая идея, особенно если сервер уже запущен и настроен, так как нам меньше всего нужно «уронить» сервер и потом пытаться перезапустить его с неправильными настройками. Для проверки сервера перед запуском можно воспользоваться параметром `-t` команды `/usr/sbin/nginx`. Она запускается с правами root:

```
nginx -t
```

Если все прошло успешно и сообщений об ошибках нет, то мы готовы к запуску сервера. Если не совсем успешно, можно посмотреть сообщения в консоли или лог-файл ошибок *Nginx*, `/var/log/nginx/error.log`. Для запуска сервера можно воспользоваться командой `service`, а выполнив сразу после нее `netstat`, можно убедиться, что мы на самом деле прослушиваем желаемый порт 80.

```
service nginx start
netstat -atnl
```

Теперь вы должны увидеть свой замечательный сайт, открыв адрес `http://127.0.0.1` в браузере на Raspberry Pi или на другом настольном компьютере. При удаленном подключении нужно знать IP-адрес системы; определить его можно, выполнив команду `ip address show` в командной строке Raspberry Pi. Эту команду можно сократить до `ip a`, если вам лень набирать лишнее. IP-адрес

указан в поле адреса inet для интерфейса eth0, если у вас проводное соединение, или wlan0, если беспроводное.

```
ip address show ##или для краткости "ip a"
```

Если вы используете Raspberry Pi в графическом режиме, установите интернет-браузер *Iceweasel*. Для нашего сайта он не понадобится, так как мы хотим, чтобы все было проще. Однако при добавлении компонентов лучше выбрать *Iceweasel* — версию *Firefox* для Debian — чем *Midori*.

```
sudo apt-get install iceweasel
```

Устанавливаем PHP

Для создания динамических страниц нужно установить PHP. С его помощью мы будем подключаться к базе данных и выполнять простые вычисления, например, преобразование температур или курсов валют. PHP очень гибок, и его установка не должна представлять сложностей, но есть одна ловушка, в которую могут попасться новички. При установке пакета `php5` в любую из систем на основе Debian пользователи часто обнаруживают, что вместе с ним был установлен web-сервер *Apache*.

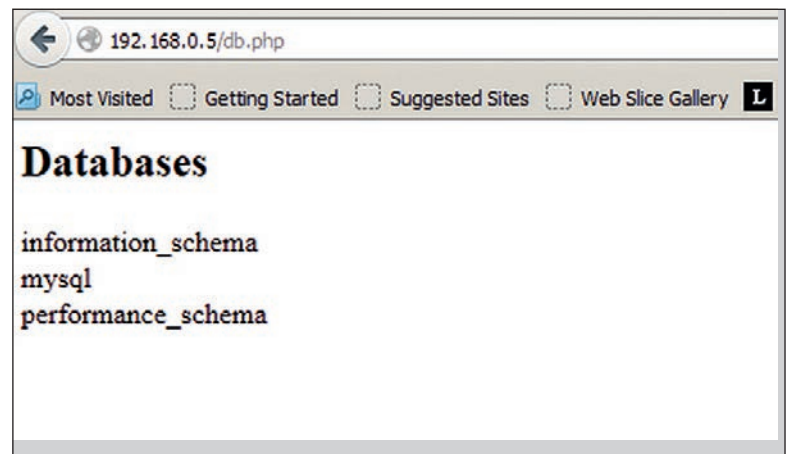
Apache является зависимостью для `php5`, но установки *Apache* можно избежать, если перед установкой `php5` установить другую зависимость из списка. Если это сделать, *Apache* не будет установлен автоматически. Для этого сначала нужно установить пакет `php5-fpm`, который позволяет PHP и *Nginx* общаться друг с другом, и избежать проблем, вызванных созданными *Apache* файлами:

```
sudo apt-get install php5-fpm php5 php5-mysql
```

Установив PHP, нужно выполнить всего одно изменение в файле `/etc/php5/fpm/php.ini`. Раскомментируйте следующие строки:

```
;cgi.fix_pathinfo=1
```

»



» С помощью функции `phpinfo()` удобно проверить работу PHP и получить конфигурацию. Проще нашей страницы не бывает, но зато как быстро мы ее создали!

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Открыв в браузере *Iceweasel* страницу <http://127.0.0.1/info.php>, мы увидим исчерпывающий список параметров PHP на Raspberry Pi.

Устанавливаем MySQL

Теперь мы можем завершить создание LEMP, установив сервер баз данных *MySQL* и выполнив его первоначальную настройку. Мы выбрали *MySQL*, так как он популярен и, следовательно, хорошо поддерживается. Существуют и другие СУБД, например, *SQLite*, но *MySQL* хорошо масштабируется, легко настраивается и может работать при большей нагрузке.

```
$ sudo apt-get install mysql-server
```

Текущая версия *MySQL* в репозиториях Debian — 5.5. Она и будет установлена. Во время установки нам предложат задать пароль *root* для *MySQL*; это не то же самое, что пароль *root* в Linux, и поэтому он хранится в базе данных учетных записей *MySQL*, а не в Linux. Выберите надежный пароль, который сможете запомнить. Пароль можно оставить пустым, но это явная угроза безопасности.

После завершения установки можно дополнительно повысить безопасность, выполнив команду `mysql_secure_installation`. Эта команда улучшит безопасность сервера *MySQL*, но добавит такие ограничения, из-за которых войти в *MySQL* от имени *root* можно будет только с локального компьютера. Оставим настройки по умолчанию без изменений. Когда все будет готово, мы сможем проверить работу сервера БД из командной строки Linux:

```
$ mysql -u root -p -e 'show databases;'
```

Мы введем имя пользователя *root*, после чего у нас спросят пароль, заданный ранее. Параметр `-e` позволяет выполнять запросы *MySQL* прямо из командной строки — наш следующий запрос выведет список всех баз данных. Разумеется, это будут системные базы данных, так как собственных мы еще не создали. Выполнив этот запрос на web-странице, можно убедиться, что *Nginx* подключается к базе данных.

Для создания демонстрационной страницы, которая будет подключаться к базе данных на PHP с сервера *Nginx*, мы создадим новую страницу PHP в `/var/www`. Откройте свой любимый

```
root@raspberrypi:/var/www# mysql -u root -p -e 'show databases;
Enter password:
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
root@raspberrypi:/var/www# █
```

➤ Запуск `mysql` в командной строке позволяет проверить правильность логина и пароля и синтаксис запроса.

редактор и создайте страницу `/var/www/db.php` со следующим содержанием:

```
<h2>Databases</h2>
<?php
$dbh=mysqli_connect("localhost","root","Password1");
$result=mysqli_query($dbh,"SHOW DATABASES");
while ($row = mysqli_fetch_assoc($result)) {
    echo $row['Database'] . "<<BR>";
}
?>
```

Опять же, мы стараемся, чтобы код оставался максимально простым. В идеальном случае логин и пароль нужно получать из другого файла, который не был бы доступен web-серверу — тогда доступ к базе данных будет только у процесса PHP. Однако сохранение кода простым удобнее для обучения.

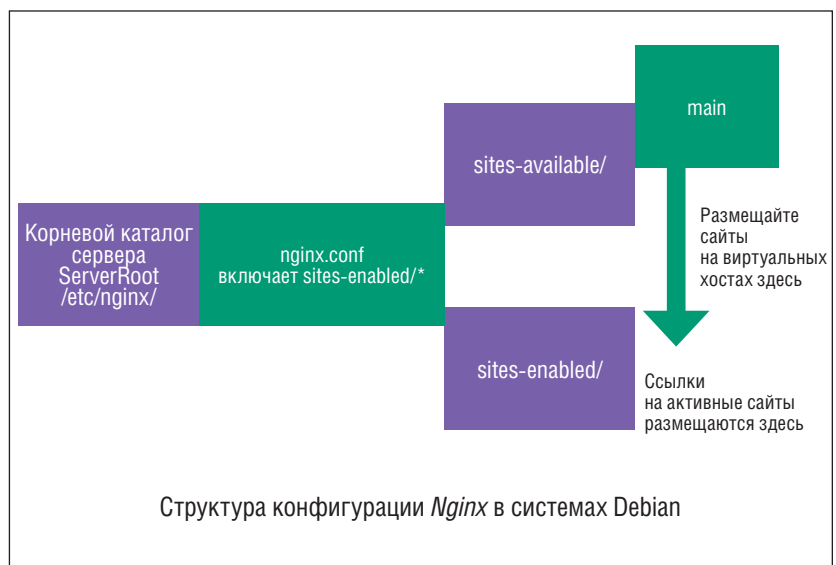
Как видите, в этом файле мы добавили к PHP-коду немного HTML-кода перед началом блока PHP. PHP-код сначала подключается к серверу *MySQL*, затем мы выполняем тот же запрос, который прежде запускали в командной строке. На этот раз результаты будут показаны в браузере — таким образом, у нас получилась простая динамическая страница, созданная из содержимого базы данных. Теперь можно порадоваться тому, что сервер LEMP настроен и запущен, и серьезно заняться разработкой на PHP и *MySQL*. **LXF**

Проверка web-сервера Nginx

В идеальном мире, прежде чем перезапускать сервер *Nginx*, мы должны проверить любые изменения, внесенные в конфигурацию. Это позволяет избежать проблем с прерыванием запуска сервера, вызванным ошибками настройки. При перезапуске сервиса нужно сначала остановить, а затем запустить *Nginx*. Остановка сервиса обычно не вызывает проблем, а перезапуск может привести к большим трудностям, если мы пропустили точку с запятой или допустили другую маленькую опечатку. Следовательно, это вызовет перебои в сервисе, пока мы будем исправлять ошибку.

После внесения любых изменений в настройки *Nginx* всегда проверяйте правильность параметров перед повторным запуском сервера. Эту проверку выполняет команда `/usr/sbin/nginx -t`, и она гарантирует, что если мы остановили сервер, то сможем запустить его снова. Такую же проверку можно выполнить командой `service nginx configtest`. Другие параметры `nginx` — `-v` (вывод версии *Nginx*) и `-V` (вывод версии и параметров конфигурации). При появлении ошибок можно заглянуть в журнал `/var/log/nginx/error.log`. Для этого часто удобна команда `tail`, так как она выводит только последние 10 строк файла:

```
# tail /var/log/nginx/error.log
```



Структура конфигурации *Nginx* в системах Debian

➤ Параметры *Nginx* и способы обращения к журналу (лог-файлу).

OpenStack: Хранилище объектов

Джолион Браун рассказывает о *Swift*, сервисе хранения объектов *OpenStack*, а затем демонстрирует управление облаком с *Heat*.



Наш эксперт

Джолион Браун

К ужасу жены, Джолион ушел со своей стабильной и уважаемой работы и начал карьеру фрилансера со специализацией на *DevOps* и *OpenStack*.



Скорая помощь

Согласованность в конечном счете — это когда копии неструктурированных данных хранятся на многих узлах, которые гарантированно вернут неизменные копии этих данных в будущем.

В августовском номере (Учебники, LXF186, стр. 76) мы продемонстрировали автоматизированную сборку облака *OpenStack* с *Puppet* и *Vagrant*. Используя эти утилиты для управления обильными настройками *OpenStack*, мы смогли создать экземпляр *Nova* для *Compute* и рассмотрели сеть *Neutron*, создав виртуальный роутер. Теперь мы воспользуемся этой работой и рассмотрим другие компоненты платформы *OpenStack*, в частности, *Swift* (модуль хранения объектов) и *Heat* (используется для управления облаком). Мы развернем небольшой кластер

Swift и разместим там несколько объектов, а с помощью текстовых файлов организуем инфраструктуру для нашего облака.

Из двух рассматриваемых проектов, *Swift* входит в состав *OpenStack* дольше. Он появился еще в 2010 году в ходе сотрудничества *RackSpace* и *NACA*, которое привело к появлению *OpenStack*. Идея *Swift* состоит в использовании недорогого «железа» для (потенциального) хранения петабайт статических неструктурированных данных — то есть данных, не хранимых в виде базы. Это делает *Swift* альтернативой прежним, очень дорогим закрытым аппаратным решениям, специализированным под конкретные задачи. *Swift* также является грубым аналогом *Amazon S3*, разделяя с ним массу характеристик, таких как согласованность в конечном счете и *http*-интерфейс *RESTful*. По *API* они также совместимы.

У *Heat* тоже *Amazon*-совместимое *API*, и он был создан с тем, чтобы наделять *OpenStack* возможностями проекта *AWS CloudFormation*. У проекта есть и собственное *API OpenStack* (и его мы рассмотрим здесь). Проект *Heat* [Жара] — названный так потому, что благодаря жаре «воспаряют облака» — интерпретирует человеко-читаемые шаблоны, которые описывают инфраструктуру облака. Затем он создает эту инфраструктуру, вызывая другие сервисы *OpenStack*.

Установка OpenStack, сжато

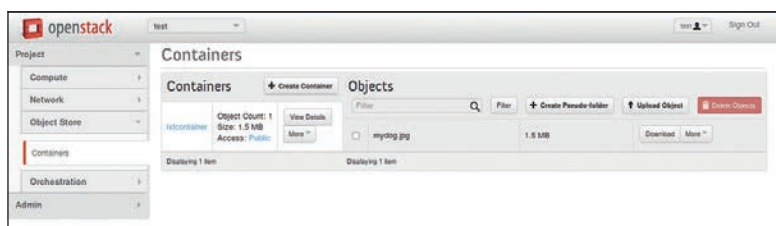
В качестве основы для этого руководства воспользуемся идеями из LXF186. Детально воспроизводить установку и настройку *Puppet* и *Vagrant* не будем: просто пробежимся по необходимым командам. Если вы хотите подробнее узнать о том, что здесь происходит, советуем почитать предыдущий учебник. Если раньше вы не пользовались *Puppet* или *Vagrant*, вы можете малость запутаться. Вкратце, *Puppet* используется для настройки *OpenStack*, а *Vagrant* — для управления виртуальными машинами (на основе *VirtualBox*), на которых работает *OpenStack*.

Для начала воспользуемся 64-битным образом *Ubuntu 14.04*, установленным на компьютер с большим объемом оперативной памяти (в нашем случае, 16 Гб). Сделав это, установим несколько необходимых программ командой *apt-get*:

```
sudo apt-get update && sudo apt-get install virtualbox git ruby-dev build-essential -y
```

После этого загрузите *Vagrant* с сайта <http://www.vagrantup.com/downloads.html>. Выберите 64-битный пакет *Linux (Deb)* и установите его (командой *dpkg -i*), затем добавьте виртуальную машину *Puppet Labs* для *Vagrant*, которую мы использовали в качестве стартовой точки для своей системы:

```
sudo dpkg -i vagrant_1.6.3_x86_64.deb
vagrant box add puppetlabs/centos-6.5-64-puppet
```



» *OpenStack Swift*: Интернету срочно нужна еще одна платформа для хранения фотографий с котятками.

Скопируйте файл `lxf-icehouse2.tar.gz` с диска этого номера в свой домашний каталог. Распакуйте его и перейдите в созданный каталог:

```
tar -zxvf lxf-icehouse2.tar.gz
cd lxf-icehouse2
```

Там есть несколько файлов, в которых прописываются сетевые параметры, в частности, `openstack.yaml` и `Vagrantfile`. Измените их согласно своим локальным настройкам; но идея здесь в том, что облако является замкнутым. Для взаимодействия с ним мы воспользуемся настольным компьютером с Ubuntu (не подключаясь к внешним системам). Для облегчения установки у нас есть небольшой скрипт — `lxfopenstack.sh`, автоматизирующий большинство действий. Чтобы его запустить, просто скомаундите `./lxfopenstack.sh`

Затем откиньтесь на спинку кресла, пока ваша старая система сыплет сообщениями о запуске облака *OpenStack*. Бегло взглянув на содержимое скрипта, вы увидите команды *Vagrant* для запуска виртуальных машин, регистрации агентов *Puppet* у мастера, подписывания сертификатов *Puppet* и установки программы *PuppetDB*. Она используется в процессе развертывания *OpenStack* под управлением *Puppet* для хранения информации о ресурсах *Swift*. Мы разворачиваем (загружаем, после чего запускаем агент *Puppet*) три узла *Swift*, а затем контроллер. Затем мы повторно запускаем агент *Puppet* на узлах *Swift*, для гарантии, что экспортированные ресурсы будут синхронизированы.

Просмотр файла `Vagrantfile` покажет, что на нашем уроке мы работаем с четырьмя узлами и мастером *Puppet*. (Попробуйте самостоятельно объединить этот файл с файлом из **LXF186**, например, добавив несколько узлов `Compute`.) Узлу `Control` в этом файле были назначены дополнительные ресурсы виртуального процессора (vCPU) и памяти, так как на нем работает большая часть сервисов *OpenStack*, включая прокси *Swift*.

Прочие узлы поддерживают оставшиеся компоненты *Swift* (сервисы хранения данных, учетных записей и контейнеров). Роли всех узлов определяются в файле `site.pp`, и мы используем `'swiftcontroller'` и `'swiftstorage'`. *Puppet* использует эти настройки для назначения ролей, заданных в модуле `puppetlabs-openstack`, устанавливаемом скриптом `puppetbootstrap.sh`. Для наших целей мы немного изменили роль `swiftcontroller`.

Во время установки могут возникать паузы. В частности, некоторое время уйдет на запуск *PuppetDB*, но потерпите, и все должно быть хорошо. Также могут появиться сообщения об ошибках, говорящие о том, что запуск агента *Puppet* уже выполняется.

```
root@control:/etc/swift
root@control:/etc/swift x vagrant@swiftstore1:~
[root@control swift]# swift-ring-builder container.builder search r1
Devices:  id region zone  ip address  port  replication ip  replication port
          0      1      3    172.16.33.10 6001    172.16.33.10 6001
          1      1      1    172.16.33.8  6001    172.16.33.8  6001
          2      1      2    172.16.33.9  6001    172.16.33.9  6001

[root@control swift]# swift-dispersion-populate
Created 26 containers for dispersion reporting, 22s, 0 retries
Created 26 objects for dispersion reporting, 0s, 0 retries
[root@control swift]# swift-dispersion-report
Queried 27 containers for dispersion reporting, 1s, 0 retries
100.00% of container copies found (81 of 81)
Sample represents 0.01% of the container partition space
Queried 26 objects for dispersion reporting, 0s, 0 retries
There were 26 partitions missing 0 copy.
100.00% of object copies found (78 of 78)
Sample represents 0.01% of the object partition space
[root@control swift]#
```

» Наша бедная перегруженная тестовая система эмулирует все эти сервисы *OpenStack* и особо не жалуется.

Не беспокойтесь — это означает, что агент, который мы пытались запустить вручную, только что запустился автоматически. Проверить состояние запуска агента *Puppet* можно, зайдя на узел (командой `vagrant ssh <имяузла>`) и заглянув в файл `/var/log/messages` на предмет сообщения “finished catalog run [запуск каталога завершился]”:

```
vagrant ssh control
sudo tail -f /var/log/messages
```

Нажмите `Ctrl+C` для выхода из команды `tail`. Когда скрипт вернет нас в командную строку, установка *OpenStack* уже готова к использованию. Команда

```
vagrant status
```

должна дать следующий результат:

```
Current machine states:
puppet                running (virtualbox)
control               running (virtualbox)
swiftstore1           running (virtualbox)
swiftstore2           running (virtualbox)
swiftstore3           running (virtualbox)
```

Web-интерфейс *OpenStack* доступен по адресу `http://192.168.11.4`. Назначьте имя пользователя `test` и пароль `abc123` (так задано в файле `openstack.yml`).

Для демонстрации основных функций *Swift* зайдите в систему через браузер. Выберите меню `Project` [Проект] в левой части экрана, а затем выберите `Object Store > Containers` [Хранилище объектов > Контейнеры]. Пункт `Create Container` [Создать контейнер] делает именно то, что должен: укажите имя контейнера и выберите



После запуска виртуальной системы в *OpenStack* в нашей тестовой среде ей может потребоваться некоторое время на достижение полной функциональности. Загляните в лог-файлы виртуальной системы, щелкнув на ее названии и перейдя на вкладку ‘Logs [Лог-файлы]’.

Swift: Курс молодого бойца

Идея *Swift* состоит в том, чтобы использовать кластеры дешевого «железа» для хранения петабайт в основном статических неструктурированных данных (то есть тех, что не организованы в базу), тем самым предлагая альтернативу обычным, очень дорогим закрытым аппаратным решениям, специализированным под конкретные задачи. *Swift* также является грубым аналогом Amazon S3, разделяя с ним массу характеристик, таких как согласованность в конечном счете и http-интерфейс RESTful.

Swift, как объясняется повсюду на нашем уроке, это широкомасштабируемая и эластичная система хранения неструктурированных данных, доступ к которым осуществляется посредством

API RESTful. Проект состоит из нескольких компонентов: прокси-серверов, обрабатывающих вызовы API; серверов, управляющих учетными записями; серверов, обрабатывающих связь контейнеров; и серверов объектов, которые работают с самими объектами. Прокси-серверы обычно находятся в системах, отдельных от остальных процессов, которые располагаются на самих узлах хранения данных. Другие фоновые задачи включают репликацию, сбор учетных записей и т.д.

Очень важная идея, используемая в *Swift* — кольца [Rings]. Они связывают сущности с их физическим расположением на диске. У учетных записей, контейнеров и объектов имеются свои

кольца, с помощью которых другие компоненты определяют их положение в кластере. Внутри колец данные можно разделить на зоны (которые могут представлять один диск, группу дисков или даже дата-центр). По умолчанию *Swift* будет всё сохранять трижды. Он попытается использовать несколько зон для достижения избыточности данных, и если зона «уйдет в оффлайн», он автоматически продублирует данные в новую зону.

Чтобы узнать подробности, взгляните на руководство администратора *OpenStack* (<http://bit.ly/OpenStackAdminGuide>). Затем прочтите руководство архитектора *Swift* (<http://bit.ly/SwiftArch>) для более глубокого ознакомления.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

режим доступа — приватный или публичный. В нашем примере мы назвали его `lxfcontainer` и сделали публичным, чтобы каждый знающий URL конкретного объекта мог его загрузить. Затем можно начать загружать в него объекты с помощью пункта меню справа. Укажите имя объекта, а потом выберите файл для загрузки.

Облачные файловые системы

Пока вам простительно не испытывать особого удивления — разве это не то же, что и хранение файлов в файловой системе? Ну, хранилище объектов выигрывает у традиционных файловых систем в стоимости, когда речь заходит о масштабируемости и избыточности данных — как только в традиционной схеме понадобится добавочное место, удобнее и легче просто бросить объекты в хранилище, чем тратить силы на настройку LUN и томов. На самом деле, при таких масштабах, которые по плечу хранилищу объектов, дешевле только лента, но у нее нет ни одного из преимуществ хранения данных в Интернете: их мгновенной доступности или физического расположения в различных местах.

Загрузить новый объект в хранилище можно через браузер или в командной строке с помощью `curl`. Кнопка `View Details` [Подробнее] рядом с контейнером сообщит нам нужный URL, нечто вроде `http://192.168.11.4:8080/v1/AUTH_...`, за которым следует длинная строка цифр и `/lxfcontainer` в конце. Для загрузки объекта к этому URL нужно добавить имя файла (в нашем случае, `/mydog.jpg`). Так как контейнер мы сделали публичным, аутентификация для загрузки не нужна.

Копнув поглубже, можно понять, как *Swift* работает на самом деле. В командной строке (в каталоге `lxf-icehouse2`) наберите

```
vagrant ssh swiftstore1
df -h
```

Тогда мы увидим, что на сервере смонтирована файловая система с `ioorback`, которая используется для хранения копий объектов. Объекты копируются между серверами *Swift* с помощью почтенной утилиты `rsync`, что и дает нам вышеупомянутую «согласованность в конечном счете». Стоит отметить, что чистая `rsync` в будущем может смениться на `ssync`, это обещает значительный прирост скорости копирования. В данном каталоге также можно воспользоваться утилитами командной строки из состава *Swift*:

```
swift --os-auth-url http://192.168.11.4:5000/v2.0/ --os-username=test \ --os-password=abc123 --os-tenantname=test list
```

Эта команда выводит список контейнеров, используемых нашим пользователем (в данном случае, всего один контейнер `lxfcontainer`). Скомандовав

```
swift --os-auth-url http://192.168.11.4:5000/v2.0/ --os-username=test \ --os-password=abc123 --os-tenantname=test stat -s lxfcontainer
```

мы получим некоторую информацию о контейнере, такую как количество объектов и размер контейнера. Для быстрого просмотра запущенных процессов *Swift* можно запустить `swift-init`:

```
sudo swift-init all status
```

Эта команда покажет другой контейнер, учетную запись и запущенные сервисы объекта. Отсутствующий здесь процесс `proxi-server` выполняется на управляющем узле.

Открыв в терминале другую вкладку (в каталоге `lxf-icehouse2`), можно обратиться к управляющему узлу и получить доступ `root` командами

```
vagrant ssh control
sudo su -
```

Заглянув в каталог `/etc/swift` на этом узле, мы увидим файлы `.builder`, примененные для создания колец (подробнее см. во врезке “*Swift*: Курсы молодого бойца”). Команда

```
swift-ring-builder container.builder search r1
```

выведет различную информацию о трех узлах *Swift*, включая IP-адреса, порты, регион/зону, в которых они находятся, и количество разделов в каждом из них. Проверить *Swift* в деле можно с помощью утилиты `swift-dispersion-populate`. Она записывает данные в некоторую часть кластера *Swift*, а результаты можно просмотреть, запустив другую команду: `swift-dispersion-report`. Чтобы эти команды работали, нужно создать файл `/etc/swift/dispersion.conf`; мы включили его в архив `lxf-icehouse2.tar`. Чтобы поместить его куда надо, зайдите в контроллер и скопируйте его из каталога `/vagrant`:

```
vagrant ssh control
sudo su -
cp /vagrant/dispersion.conf /etc/swift
```

Заглянув в этот файл, мы увидим, что параметр `dispersion_coverage` установлен в `0.01` — это доля кластера *Swift*, которую будет использовать команда. Перед запуском функции заполнения попробуйте изменить это значение, но будьте осторожны! Мы во время тестирования переполнили свою маленькую систему. Конечно, в нашем случае хорошо то, что благодаря автоматизированной установке можно легко удалить всю систему командой `vagrant destroy -f` и собрать ее снова скриптом `linuxformat.sh`.

О *Swift* можно рассказать еще много интересного; дополнительную информацию ищите на <http://docs.openstack.org>.

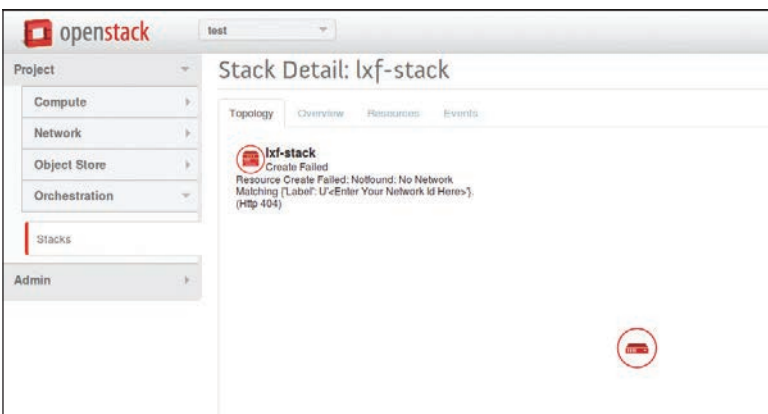
Нагреваем облака

Двинемся дальше и рассмотрим *Heat*, модуль управления *OpenStack*. Соответствующий пункт меню в панели управления находится прямо под хранилищем объектов. Если щелкнуть на подмену `Stacks` [Стеки], откроется список — к сожалению, пока пустой. Исправим это недоразумение, запустив стек прямо сейчас с помощью ссылки справа. В следующем окне можно выбрать шаблон. Выберите `File` [Файл] из первого выпадающего меню, затем зайдите в каталог `lxf-icehouse2` и выберите файл `lxf-simple.template`. Нажмите `Next` [Далее], укажите имя стека и введите пароль пользователя `test` (`abc123`) в соответствующее поле. Теперь нажмите `Launch` [Запуск], откиньтесь на спинку кресла и созерцайте, как ваша инфраструктура собирает себя, подобно сцене из очередного ужасного сиквела «Трансформеров». Секундой позже процесс сборки остановится. Щелкните на названии стека, чтобы узнать, что случилось.

Поскольку по умолчанию в нашем облаке *OpenStack* доступны несколько сетей, в нашем шаблоне нужно указать правильный идентификатор сети. Выберите меню `Network` [Сеть] в правом



В Интернете много шаблонов для AWS CloudFormation, и они должны работать в *OpenStack* — с малыми изменениями или даже без них.



➤ Изменение шаблонов может оказаться работой с полной занятостью.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Heat: Курс молодого бойца

Heat — сервис управления облаком на основе шаблонов, изначально основанный на (и совместимый с) AWS CloudFormation. Однако *Heat* получил собственное развитие и создал свой формат шаблона — Heat Orchestration Template (HOT). Подробную информацию можно найти в руководстве *OpenStack* по HOT: <http://bit.ly/HotGuide>. HOT — это шаблон на основе YAML в человеко-читаемом формате.

Шаблоны *Heat* состоят из инструкций по построению облака или инфраструктуры путем вызова других компонентов системы *OpenStack*, которые создают ресурсы (экземпляры систем, IP-адреса,

тома и т.д.). Кроме того, в *Heat* можно создавать высокодоступные экземпляры систем, и имеется автомасштабирование, т.е. системы могут изменять свой размер по запросу пользователя. Этот сервис связан с Ceilometer, модулем *OpenStack*, собирающим информацию об использовании системы. *Heat* поддерживает связи между ресурсами, что позволяет строить инфраструктуру в правильной последовательности.

Шаблоны также можно обновлять и использовать для изменения существующего стека. После окончания работы *Heat* удалит всю инфраструктуру,

определенную в простом текстовом файле.

На уровне операционной системы *Heat* состоит из ряда различных компонентов — клиента командной строки, сервиса REST API, AWS-совместимого API и самого модуля *Heat*.

Heat постоянно развивается, но есть ресурсы, на которые стоит взглянуть, чтобы узнать о нем больше. На <http://docs.openstack.org> есть документация, в GitHub — <https://github.com/openstack/heat-templates> — несколько репозиторий с примерами шаблонов, а на <https://github.com/rackerlabs/heat-tutorial> — углубленное руководство от Rackspace.

верхнем углу экрана, затем Networks [Сети], и вы увидите нашу публичную и приватную сети. Щелкните на Private [Приватная], чтобы получить информацию о ней, и найдите идентификатор (длинная последовательность букв и цифр), используемый *OpenStack* для обращения к этой сети. Выделите его и скопируйте в буфер обмена. С помощью любого подходящего редактора откройте файл `lxf-simple.template` в каталоге `lxf-icehouse2` и замените сообщение 'Enter your network ID here [Введите сюда идентификатор своей сети]' содержимым буфера обмена. Сохраните файл, вернитесь в меню Orchestration > Stacks [Управление > Сетки] и удалите стек, затем запустите другой.

Используйте те же настройки, что и в прошлый раз, и — теперь через некоторое время — ваш стек будет успешно создан. Конечно, это очень простой шаблон, состоящий из одного экземпляра системы Cirros, подключенного к сети. Прежде чем к нему подключаться, нужно назначить ему плавающий (публичный) IP-адрес. Для этого используется меню Compute > Instances [Вычисления > Системы]. Также нужно разрешить кое-какой дополнительный доступ — сделайте это в меню Compute > Access [Вычисления > Доступ] и Security > Security Groups [Безопасность > Группы безопасности]. Откройте правила группы по умолчанию и воспользуйтесь меню Add Rule [Добавить правило], чтобы добавить правила для SSH, ALL ICMP и HTTP. Теперь вы сможете подключиться к системе по ssh со своего рабочего стола как пользователь Cirros с паролем `cubswin:`. Превосходно. Теперь попробуем кое-что посерьезнее.

Запустите новый стек и на сей раз выберите файл `lxfwordpress.template`. После нажатия Next станет ясно, что настройки этого стека чуть сложнее. Шаблоны *Heat* в большой степени можно разбить на параметры. Здесь есть параметр `image_id`; наведя на него мышью, мы увидим подсказку (из файла шаблона). Здесь рекомендуется использовать образ Fedora, поэтому быстро перейдем к его загрузке. Щелкните по меню Admin [Администрирование] слева, выберите Images [Образы] и создайте образ. Назовите его `fedora-20.x86_64`, выберите Image Location [Размещение образа] в окне Image Source [Исходный образ] и введите URL http://download.fedoraproject.org/pub/fedora/linux/updates/20/Images/x86_64/Fedorax86_64-20-20140407-sda.qcow2 в поле Image Location [Размещение образа]. Формат нашего образа — QCOW2. При желании сделайте образ публичным, щелкнув на соответствующей опции. Наконец, нажмите Create Image [Создать образ]. Образ довольно большой, поэтому в течение некоторого времени, которое зависит от ширины канала, доступного нашей системе, он будет находиться в состояниях Queued [В очереди] и Saving [Сохранение]. Пока все это происходит в фоне, воспользуемся моментом и создадим ключевую пару SSH для использования с созданной системой. Вернитесь в меню Project [Проект] и в разделе Compute [Вычисление] выберите Access & Security

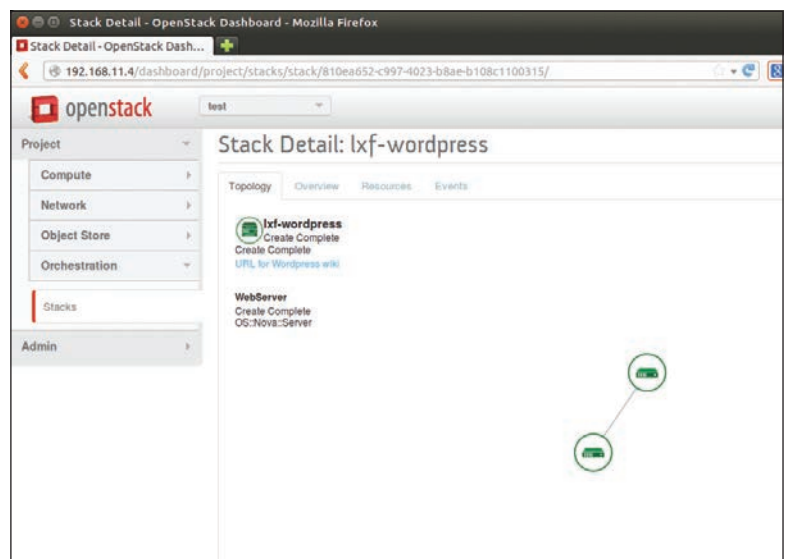
[Доступ и безопасность], затем перейдите на вкладку Key Pairs [Ключевые пары]. Создайте ключевую пару и назовите ее "LXF". При появлении соответствующего сообщения сохраните полученный файл `.pem`.

Наконец, нужно изменить файл `lxf-wordpress.template`, добавив в него идентификатор нашей сети. Сделайте это так же, как ранее с файлом `lxfsimple.template` (замените строку «Введите сюда идентификатор своей сети» на идентификатор сети).

Блоггинг на основе облака

Пора нашему образу Fedora загрузиться, а его состоянию в меню Images — измениться на активное. Если да, вернитесь в Orchestration/Stacks, запустите новый стек и снова выберите файл `lxf-wordpress.template`. Укажите имя стека, пароль тестового пользователя (`abc123`) и "LXF" в поле с ключевой парой. Стек создается довольно долго; это время зависит от размера запускаемой настольной системы. В конце концов вы будете вознаграждены статусом «Завершено», и появятся два компонента: web-сервер и база данных. Они образуют маленькую демо-установку *WordPress*. Назначьте web-серверу публичный IP-адрес и откройте его в браузере, чтобы увидеть результат.

Надеюсь, что наше руководство наглядно продемонстрировало основные возможности *Heat* — просмотрите более сложные примеры, которые можно найти в Интернете, и попробуйте включить их в свою новехонькую систему *OpenStack* под управлением *Heat*. **LXF**



► Наконец все получилось! Наша система из двух узлов готова.



PHP/SQLite: Почта чохом

Возвестите всему миру! **Кент Ельчук** учит делать массовые новостные рассылки с вашего сервера, обходя большинство фильтров спама.



Наш эксперт

Кент Ельчук — web-разработчик в Ванкувере, Канада. В свободное время он взахлеб пишет приложения, блоги и книги и экспериментирует с VM и Raspberry Pis.

Тема почтовых рассылок порождает как положительные, так и отрицательные эмоции. Если рассматривать ее с одной колокольни, то рассылки позволяют вам поддерживать связь со всеми своими клиентами и уведомлять их обо всех предложениях; а с другой, на более персональном уровне — понятна ярость, которую они вызывают, заполняя ваш почтовый ящик хламом, который вам совсем не нужен.

Но отложим эмоции в сторону: наш урок посвящен организации массовой рассылки со своего сервера, независимо от того, используете ли вы отдельный компьютер от своего хостинг-провайдера, Raspberry Pi, или Linux на USB-брелке.

И хотя это приложение сработает на любом устройстве, где стоит Linux, наилучшая опция — использовать его на своем веб-хосте, поскольку оно сможет обойти более сильные спам-фильтры. Если вы работаете с приложением на хосте, где IP-адрес не принадлежит к тому же домену, он может быть не принят некоторыми получателями. Но, поскольку для аутентификации отправителя электронной почты оно использует SMTP, то вы уже обошли критерии большинства фильтров спама.

Планируя разместить сайт у себя дома на компьютере с Linux, будьте осторожны при массовой рассылке по электронной почте, чтобы ваш IP-адрес не угодили в черный список. Можете проверить свой домен и IP на предмет нахождения в черном списке на www.mxtoolbox.com. Если он окажется в черном списке mxtoolbox, просить удалить его оттуда придется ну очень вежливо. Возможно, вы также захотите получить новый IP-адрес от своего интернет-провайдера.

Помня все это, можете смело знакомиться со следующим набором инструкций по настройке собственного скрипта массовой рассылки по электронной почте. Поскольку используемый в данном примере код написан на PHP и MySQL, вам нужно, чтобы они оба были установлены на вашем компьютере.

Предпосылки установки

Чтобы этот скрипт заработал, нужно запустить последовательность разных команд. Первая команда установит на ваш компьютер с Linux SQLite. Установка SQLite в Linux означает, что вы всегда сможете получить доступ и изменить файл базы данных SQLite в файловой системе.

В отличие от MySQL, база данных SQLite является обычным файлом, и, следовательно, вам нужно только открыть файл с помощью SQLite, чтобы выбрать данные или внести изменения. Команда для установки SQLite показана ниже:

```
root@raspberrypi:~# apt-get update
root@raspberrypi:~# apt-get install sqlite3
```

Установив SQLite в Linux, надо установить пакеты Pear, используемые для рассылки сообщений электронной почты через SMTP. Следующая серия команд показана ниже:

```
root@raspberrypi:~# apt-get install php-pear
root@raspberrypi:~# pear install Mail Mail_Mime
root@raspberrypi:~# pear install Net_Smtp
```

И, наконец, библиотека SQLite добавляется к PHP, чтобы вы могли использовать базу данных SQLite:

```
root@raspberrypi:~# apt-get install php5-sqlite
```

После установки модуля *php sqlite*, Apache перезапустится. Теперь вы готовы к соединению с базами данных SQLite.

Остальная часть этого урока будет посвящена рассылке индивидуальных сообщений по электронной почте на адреса вашего индивидуального списка рассылки. Есть всего несколько небольших файлов, которые позволяют рассылать электронную почту, изменять шаблоны и отслеживать информацию от получателей.

Настройка массовой рассылки

Главный файл, который берет на себя львиную долю работы — **index.php**. Этот файл будет отправлять нужный шаблон сообщения нужной группе людей. Помимо отправки сообщений, на этой странице есть ссылки, которые позволяют вам редактировать шаблоны. Поскольку у вас могут быть разные группы для рассылки, было бы нормально иметь несколько шаблонов, пригодных для специальных сообщений или для периодической новостной рассылки.

Строки с третьей по пятую в файле **index.php** устанавливают соединение с базой данных и включают файлы заголовков, используемые для навигации. Возвращается объект \$dbh и используется для всех запросов базы данных SQLite внутри приложения массовой рассылки по электронной почте. Запросы базы данных будут делаться при открытии страницы, и еще больше запросов будут выполнены при нажатии на кнопку Submit и отправке электронной почты.

Поскольку вы не нажали на кнопку отправки массовой рассылки, вы увидите страницу, где перечислены ваши опции рассылки; поле ввода и выпадающие меню. Весь код HTML, который вы здесь увидите, из любого результата, который не подпадает под условие `if ($_POST['submit']) { }`.

Результат и исполнение кода, которое вы видите в ранее описанном условии, происходит только тогда, когда форма отправляется.

Чтобы вышеописанное стало немного понятнее, строки кода 1–9 и строки от 99-й до конца файла отображаются, когда вы не отправляете форму. Поле ввода можно использовать для добавления индивидуальной темы сообщения, например, «Привет от Кента».

Первое выпадающее меню можно использовать для выбора группы, которая будет получать рассылку. Получатели рассылки размещаются в таблице контактов. Помимо прочей информации,

Скорая помощь



База данных SQLite имеет расширение по умолчанию **.sqlite**, которое вы при желании можете переименовать в **.db**. Также ей нужно будет настроить разрешения на 777, если вы планируете изменять ее из своего приложения. В противном случае вы получите только частичные привилегии.

таблица содержит уникальный ID, имя, фамилию, адрес электронной почты, номер телефона, категорию и дату каждого контакта. Поскольку каждый контакт имеет адрес электронной почты и ID соответствующей категории, сообщение может быть доставлено. Категории контактов размещаются в таблице **leads_categories**.

Второе выпадающее меню можно использовать для выбора шаблона, который вы хотите отправить. Таблица, где хранятся эти данные, называется **leads_email_templates**. Эта таблица содержит столбец для id шаблона, имени шаблона и кода шаблона. Настоящий файл шаблона — код HTML.

Ваш шаблон может быть чем угодно, от простого текста до коммерческого адаптивного шаблона. Чистые адаптивные шаблоны, которые отлично выглядят на всех настольных ПК и мобильных устройствах, можно приобрести за несколько фунтов/долларов, и в одном приобретенном шаблоне вы нередко получаете несколько вариантов раскладки.

Альтернатива — взять исходный код из полученного вами сообщения электронной почты и добавить туда что-то свое.

Независимо от того, где вы используете свой шаблон, вы можете соединить ссылкой ваши изображения с файлами на удаленном хосте или встроить в них data:urls. Встроенное изображение будет использовать тэг src, который выглядит, как следующий пример:

```

```

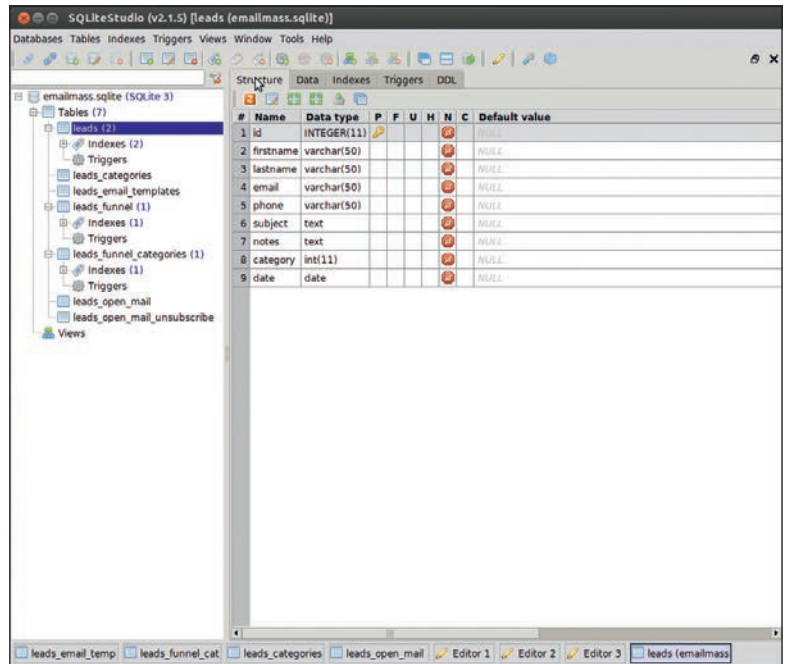
После кнопки Submit

Когда вы нажмете на кнопку Submit, вся страница кода является результатом, потому что нет никаких условий, не разрешающих ей становиться таковым. Поскольку код, который вы видите перед тем, как форма будет отправлена, описывался в предыдущем разделе, этот раздел руководства только детализирует реальный код, исполняемый после того, как форма будет отправлена.

Для начинающих, перейдите на строку 10. Поскольку переменная '\$_POST['submit']' настроена, первый запрос запускает и возвращает все контакты из базы данных для выбранной категории. Все строки, удовлетворяющие этому критерию, собираются, и каждая строка включается в массив под названием **ids_array** и массив под названием **\$mails_array** с помощью цикла foreach. Выражение **print_r(\$mails_array)** после цикла foreach выводит каждый адрес электронной почты, который был добавлен в **\$mails_array**.

Следующее условное выражение — `if (ISSET($mails_array)) {}` используется для отправки сообщений, если в **\$mails_array** нет адресов электронной почты. Если адресов электронной почты нет, код не исполняется, и вашему экрану не приходится показывать вам все эти досадные сообщения об ошибках.

Далее мы предполагаем, что у вас есть получатели, которым будут отправлены сообщения. Итак, чтобы отправлять сообщения, по одному за раз каждому получателю, используется цикл



foreach для отправки сообщений каждому человеку в **\$mails_array**. Код **foreach (\$mails_array as \$key => \$myemail)** создаст **\$key** и значение **\$myemail** для каждого человека. Значение **\$myemail** — это каждый реальный адрес электронной почты получателя.

Внутри цикла foreach основная работа, которую выполняет код, заключается в создании темы. Если вы добавили личное сообщение, пользователь получит его. Если вы личного сообщения не добавили, вместо него используется сообщение по умолчанию **My Business Message**.

Когда тема создана, шаблон берется из таблицы базы данных под названием **leads_email_templates**. В этой таблице три столбца, поэтому вы можете разбить шаблон на заголовок, тело и подпись [footer]. Или можете просто использовать один столбец для всего шаблона.

Шаблон выходит из базы данных в качестве строковой переменной **\$messagea** и конвертируется в переменную **\$body**. Сразу же после создания переменной **\$body** создаются заголовки from [от кого], to [кому] и subject [тема]. Они создадут переменную **\$headers**, отправляемую вместе с электронной почтой. Следующая строка кода — настройки аутентификации SMTP электронной почты. Они будут содержать отправителя, адрес хоста (т.е. `ssl://smtp.gmail.com`), имя пользователя и пароль.

В этом примере сообщение отправляется через SMTP. Вкратце это означает, что вы аутентифицируетесь через реальную учетную запись электронной почты, например, в Gmail. Фактически, Gmail подходит идеально, поскольку вы аутентифицируетесь через соединение SSL через порт 465.

» Вкладка Structure показывает информацию для вашей таблицы, например, имя и тип. Здесь можно внести изменения и применить их.

»

Соединения SQLite и MySQL

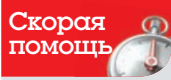
Эту функцию соединения можно использовать для баз данных SQLite или MySQL. Просто раскомментируйте строки для той \$dbh, с которой вы намерены работать.

```
function db_connect()
{
```

```
global $dbh;
$user = 'root';
$pass = '';
// MYSQL CONNECTION
// $dbh = new PDO("mysql:host=localhost;db
name=email_mass", $user, $pass);
```

```
// SQLITE CONNECTION
$dbh = new PDO('sqlite:db/emailmass.sqlite');
$dbh->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
return $dbh;
}
```

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



Скорая помощь

Можно конвертировать базу данных MySQL в SQLite одной командой:

```
root# ./mysql2sqlite.sh
-u username -p
MyPassword database_from sqlite3
database_to.sqlite
```

Сразу же после того, как вы настроите свои регистрационные данные, метод `$smtp->send()` доставит почту. Если доставка пройдет успешно, вы увидите: “Mail to: example@example.com has been successfully sent! [Сообщение успешно доставлено]”, где example@example.com — один из адресов из базы данных. Если в вашей таблице базы данных нет адресов на момент запуска электронной почты, вы увидите сообщение: “No emails exist [Адресов не существует]”.

Файлы шаблона

Главный файл, для которого вы редактируете свои шаблоны — **viewtemplate-code.php**. Доступ к этой странице получают посредством щелчка по ссылке Edit на основной странице. Открыв эту страницу, вы можете редактировать и сохранять свой файл шаблона.

Файлы шаблона загружают данные, которые хранятся в таблице базы данных **leads_email_templates**. Когда вы щелкаете по кнопке Submit, чтобы отправить сообщение со страницы **index.php**, данные собираются для выбранного шаблона. При редактировании к шаблону добавляются обратные слэши, поскольку он вставляется в базу данных. Когда шаблон считывается для редактирования или используется по назначению, слэши удаляются, и, будучи доставлен получателю, шаблон имеет вполне достойный вид. Если вы хотите просмотреть свои файлы шаблона перед отправкой, у вас есть три опции.

Первая — создать тестовую категорию (или использовать категорию Test из таблицы **leads_categories**, которая включена в файлы кода). Хотя эта опция весьма важна для тестирования ваших сообщений перед их отправкой подписчикам, все же это может оказаться слишком утомительным и затратным по времени методом создания идеального шаблона; особенно если это коммерческий шаблон с адаптивным дизайном.

Если так вышло, что вы используете шаблон HTML, вашей второй опцией будет использовать редактор для кодирования шаблона. Чтобы протестировать шаблон, вы просмотрите его в браузере, как любую web-страницу. Получив удовлетворительный результат, можете обновить существующий шаблон или создать новый.

Последняя опция — использовать простой встроенный редактор. Вы можете получить доступ к редактору, открыв файл **index.php** и щелкнув по имени темы. Это приведет к файлу **view-template.php** и выведет вместе с ним id шаблона. Например, вы будете перенаправлены на страницу, которая выглядит так: **view-template.php?templateid=1**.

Итак, открыв **index.php**, в строке 168 вы увидите ссылку, как в следующем примере:

```
<a href = "view-template.php?templateid=' . urlencode($id_template) . "'>
```

Ссылка откроет страницу **view-template.php**, а вопросительный знак (?) добавит параметр **templateid**, содержащий значение для id из таблицы **leads_email_templates** в базе данных SQLite. Неплохо будет использовать функцию **urlencode()**, особенно если вы используете длинные URL со множеством параметров. Вы можете удалить эту функцию, но она все равно будет работать.

Когда откроется файл **view-template.php**, он будет включать файл соединения с базой данных и файл заголовка. Затем URL будет декодирован с помощью функции **urldecode()**, и все символы подвергнутся обработке функцией **htmlentities()**. Поскольку это только число из базы данных, она будет одна.

В данном случае применение функций **urlencode()**, **urldecode()** и **htmlentities()** может показаться перебором. Но всегда стоит подумать о безопасности: web-приложения часто взламывают или используют злонамеренно, поскольку пользователь может начать развлекаться со строковыми переменными URL, вызывая проблемы. Намного чаще это происходит, когда злоумышленник точно знает про уязвимость.

Идем дальше: функция **htmlentities()** и константа ENT_QUOTES используются, чтобы обеспечить большую безопасность вашего приложения. И снова, это не обязательно, но хорошей привычкой будет проверять входящую информацию перед использованием ее в запросе или элементе формы.

Затем следующие строки кода берут данные, существующие в столбцах заголовка, тела и футера для нужного вам шаблона. После получения данных шаблон отображается на вашей странице.

Отслеживание полученных сообщений

Теперь, заимев портативную программу массовой рассылки, вы можете удовлетвориться использованием ее исключительно для отправки сообщений своим получателям. Однако вам, возможно, захочется отслеживать, кто из ваших получателей открыл ваши сообщения. Для этого примеры кода предлагают два файла: **track-email.php** и **leads-opened-mail.php**.

Файл **track-email.php** активируется, когда пользователь решает вашему изображению отображаться в шаблоне HTML. Когда ссылка возвращается на вашу страницу, база данных обновляется.

Если вы планируете отслеживать пользователей, можете запускать это приложение на сервере дома или на учетной записи web-хостинга. Если ваш хостинг — дома, проверьте, разрешили ли вы роутеру пересылать порт 80, чтобы клиентская ссылка соединилась с вашими файлами.

Чтобы отслеживать полученное сообщение, вы создаете сообщение со скрытой ссылкой на изображение, которая указывает на файл PHP. Код можно добавить в файл **index.php** под переменной **\$footer** около строки 53. Что будет выглядеть примерно так:

```
$footer .= '<img src=>http://example.com/email-mass/track-email.php?id=' . $id_of_lead . '>';
```

Это уже работает, но вы можете шагнуть дальше и найти временную адреса электронной почты, чтобы убедиться, что удален реальный член списка, а не тот, кто создает URL с разными id, что может вызвать появление неуместных пунктов. Файл **leads-opened-mail.php** предоставит вам список всех тех, кто открывал сообщение с изображением. Найти это можно в основном меню.

Вы также должны предоставить своим подписчикам возможность отказаться от получения. Для этого можно использовать таблицу **leads_open_mail_unsubscribe**.

Поскольку это — руководство, вы можете скопировать код с **track-email.php** и назвать его как-нибудь вроде **unsubscribe.php**.

Предоставьте SMTP данные об электронной почте

Чтобы программа отправки SMTP заработала, нужно предоставить ваш хост электронной почты, порт, имя пользователя и пароль. Приведенный ниже пример будет

работать для любой учетной записи Gmail, если подставить верные хост, имя пользователя и пароль.

```
$from = "John Do";
```

```
$host = "ssl://smtp.example.com";
$port = „465“;
$username = „example@example.com“;
$password = «<passwordhere>»;
```


Оба будут использовать id пользователя в ссылке. Единственное различие в том, что вы вводите данные в таблицу `leads_open_mail_unsubscribe`, которая включена в базу данных.

Чтобы получатель получил ссылку для отказа от рассылки, вы добавляете следующую строку под переменной `$footer`:

```
$footer .= '<a style="text-decoration:underline; color:#f16423;" href="http://example.com/email-mass/unsubscribe.php?id='.$_SESSION['id_of_lead'].'">Unsubscribe</a>'
```

Если кому интересно — “.” просто прилагает ссылку к переменной `$footer`.

Хотя наш урок мог быть написан и для *MySQL*, по переносимости база данных *SQLite* является более удачной опцией. Если вы возьмете в качестве базы данных *MySQL*, вам может понадобиться настроить новую базу данных по каждому месту, для которого вы размещаете файлы.

С помощью нашей инструкции вам всего лишь нужно скопировать папку для этого приложения и вставить ее на любой сервер, на котором работает PHP. Основной фактор использования базы данных *SQLite* в том, что она должна быть записываемой. Поэтому и папке `db`, и файлу базы данных *SQLite* выдается разрешение `777`. Это делает команда `chmod 777`. Если разрешение не `777`, вы сможете выбирать из базы данных, но не сможете внести изменения с помощью команд `insert`, `update` и `delete`.

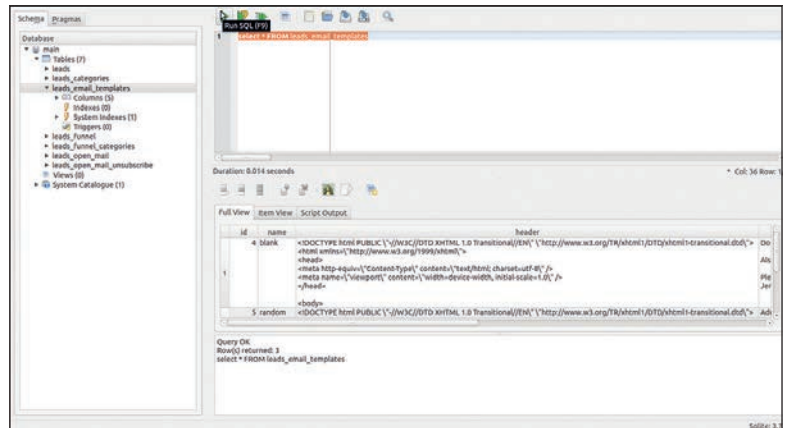
Если вы хотите работать с *SQLite* из командной строки, такое тоже возможно. Примеры для командной строки имеются. Получив доступ к базе данных *SQLite*, вы сможете запустить любую из желаемых команд *SQLite*. Полный список приведен на <http://sqlitestudio.pl>.

Желая запустить *SQLite* через GUI, лучше всего установить на вашей локальной системе Linux инструмент, подобный *SQLite Studio* или *SQLiteMan*. Оба они бесплатны и просты в использовании. *SQLiteMan* заработает сразу, а вот *SQLite Studio* нужно скачать и сделать исполняемым (<http://sqlitestudio.pl/?act=download>).

```
Например, после скачивания в Ubuntu вы запускаете
chmod +x sqlitestudio-2.1.5.bin
./sqlitestudio-2.1.5.bin.
```

Откроется ваш инструмент *SQLite Studio*. Вообще говоря, оба инструмента GUI *SQLite* делают примерно одно и то же, позволяя читать и обновлять таблицы и создавать индивидуальные запросы. На персональном уровне, *SQLite Studio* выглядит лучше и имеет больше функций. В отличие от использования для управления базами данных на вашем веб-сервере встроенного инструмента администрирования типа *phpMyAdmin*, ваше редактирование *SQLite* происходит на локальной машине. Потом вы просто загружаете его с помощью *SFTP* после внесения изменений. Если вы используете такую IDE, как *PHPStorm*, *Netbeans* или *Eclipse*, можете открыть свой проект и переместить файлы на свой сторонний сервер почти мгновенно. Поэтому если обнаружится, что вы работаете с приложением, которое использует базу данных *SQLite*, у вас будут одновременно открыты инструмент *SQLite* и Coding IDE.

Чтобы открыть файл своей базы данных, нужно перейти в его папку с помощью командной строки и открыть ее командой



SQLite3. Три строки ниже покажут вам, как открыть вашу базу данных и просмотреть все таблицы. Теперь вы получили полную свободу действий в своей базе данных *SQLite*.

```
root@raspberrypi# cd /var/www/email-mass
root@raspberrypi# sqlite3 emailmass.sqlite
sqlite> .table
```

Две следующих строки показывают простое утверждение `'SELECT'` и `'DELETE'`.

```
sqlite>SELECT * FROM leads_email_templates WHERE id =3;
sqlite>DELETE FROM leads_email_templates WHERE id =1;
```

SQLite Studio скачивается бесплатно. Скачав и открыв его, вы получите полный контроль над своей базой данных.

Чтобы открыть базу данных с помощью *SQLite Studio*, выберите в основном меню Databases, затем перейдите в Add Database > Find It > OK. В следующий раз, когда вы откроете *SQLite Studio*, загрузится также и эта база данных.

Чтобы поработать с таблицей базы данных, дважды щелкните по одной из таблиц. В этот момент вам нужно будет использовать вкладки Structure и Data для индивидуальной настройки и изменения в записях. Щелкнув по значку Карандаш, вы сможете написать индивидуальный запрос. Затем, если щелкнуть по значку с молнией, команда выполнится.

Все приложение целиком можно использовать с обычной базой данных *MySQL*. Единственное, что нужно реализовать — это превращение базы данных в базу данных *MySQL* и изменение соединения с файла `public_DBO.inc`. Код уже имеется; просто раскомментируйте соединение *MySQL* и прокомментируйте соединение *SQLite*. После добавления правильного имени пользователя и пароля все будет готово. В своем нынешнем виде оно настроено на пользователя root без пароля.

Это небольшое приложение массовой рассылки по электронной почте позволяет легко создать и разослать сообщение за считанные минуты. Оно может работать на любом локальном хосте, с которым вы работаете. Вы можете использовать Raspberry Pi, USB-брелок, виртуальную машину или учетную запись своего веб-хостинга. **LXF**

» Вот что вы увидите, запустив *SQLiteMan*. Можете просматривать содержимое в любой таблице и запускать индивидуальные запросы.



Когда ваш первый столбец обрабатывается *MySQL* с автотипом, обычно он имеет тип `Int` и первичный ключ. В *SQLite* вам нужно конвертировать тип `Int` в `INTEGER` и сделать его первичным ключом. По умолчанию конвертирование базы данных *MySQL* с таблицами, которые используют тип `int`, произойдет в тип `Int` в *SQLite*.

Функция SMTP vs Mail()

На нашем уроке мы использовали классы почты `PEAR` и рассылали сообщения через SMTP. Однако вы можете использовать встроенную функцию PHP `mail()` в файле `index.php`. Если используется функция `mail()`, вероятно, что она вернет заголовки `X-Source` в сообщении электронной почты, что отследит его вплоть до скрипта,

с которого оно было отправлено. Заголовки показаны ниже. Использовать функцию `mail()` проще, но не лучше:

```
X-Source: /usr/bin/php
X-Source-Args: /usr/bin/php /home/username/public_html/contact.php
X-Source-Dir: example.com:/public_html
```

В примере с кодом строка, которая управляет сообщением, вот:

```
$mail = $smtp->send($myemail, $headers, $body);
```

Если вы хотите использовать функцию `mail()`, следующий код выполнит этот трюк.

```
mail('username@example.com', 'My Email Subject', $body);
```

Ядро Linux: Глянем внутрь

Дмитрий Кузнецов проникает в святая святых операционной системы Linux и обнаруживает, что это всего лишь программа.



Наш эксперт

Дмитрий Кузнецов
Более 10 лет ведет неравный бой с монстрами сложности, хотя некоторые почему-то их считают ветряными мельницами.

Разработка операционных систем (ОС) — одна из интереснейших областей программирования. К сожалению, ее принято считать уделом «заморских бородатых гиков», а потому далеко не все решаются заглянуть «под капот» своей системы. Конечно, написать ОС с нуля — задача не из легких. Однако с приходом Linux, исходный код ядра которой открыт, появился не столь радикальный путь познания. Теперь можно начать

с небольших экспериментов над уже готовой системой. Но даже здесь иногда бывают затруднения. Они, безусловно, не так значительны, но все же часто имеют место, а потому вполне достойны внимания. В данной статье на примере Ubuntu 12.04 LTS 64-bit рассматриваются два, вероятно, наиболее важных из них: безопасное тестирование модифицированного ядра Linux и специфичность методов его отладки.

ГОТОВИМ «ИСПЫТАТЕЛЬНЫЙ ПОЛИГОН»

Сборка ядра Linux

Первым делом нужно обзавестись объектом для экспериментов — собрать ядро Linux из исходного кода. Его архив проще всего получить из репозитория:

```
apt-get install linux-source
```

Создадим рабочий каталог, скопируем туда полученный архив и распакуем его:

```
mkdir /home/kernel_test
```

```
cd /home/kernel_test
```

```
cp /usr/src/linux-source-3.2.0/linux-source-3.2.0.tar.bz2 ./
```

```
tar -xjvf ./linux-source-3.2.0.tar.bz2
```

```
cd ./linux-source-3.2.0
```

Собирать будем с текущей системной конфигурацией, которую можно найти в каталоге `/boot`. Применить ее к нашему чистому дереву исходного кода можно так:

```
make oldconfig
```

Запустим систему конфигурирования:

```
make menuconfig
```

Менять ничего не будем, лишь параметру General setup > Local version — append to kernel release присвоим значение `-test`. Оно

добавится к версии ядра, что позволит отличать его от других. Дважды нажав Esc и согласившись с запросом о сохранении изменений, запускаем сборку:

```
make
```

Результаты сборки удобно держать в отдельном каталоге. Создадим его:

```
mkdir ../result
```

Образ ядра копируем вручную:

```
cp ./arch/x86/boot/bzImage ../result/
```

Для модулей (файлы `*.ko`) и прошивок (firmware; это небольшие двоичные файлы, необходимые некоторым устройствам для корректной работы и хранящиеся в подкаталоге `firmware` дерева исходного кода ядра в готовом виде) есть специальная команда: `INSTALL_MOD_PATH="../result/_mods" make modules_install`

Учтите: в системе также могут присутствовать сторонние модули (например, драйверы устройств, не входящие в ядро). В этом случае каталог `/lib/modules/<текущая версия ядра>`, кроме `kernel`, должен содержать еще и другие подкаталоги с `ko`-файлами. Их также следует скопировать в `../result/_mods/<версия ядра>`, чтобы наше тестовое ядро могло их найти.

Для контроля можно посмотреть версию собранного ядра и удостовериться, что `-test` к ней действительно добавилась:

```
make kernelrelease
```

Остается сгенерировать зависимости, чтобы ядро могло ориентироваться в своих модулях (3.2.55 — версия ядра на момент написания статьи):

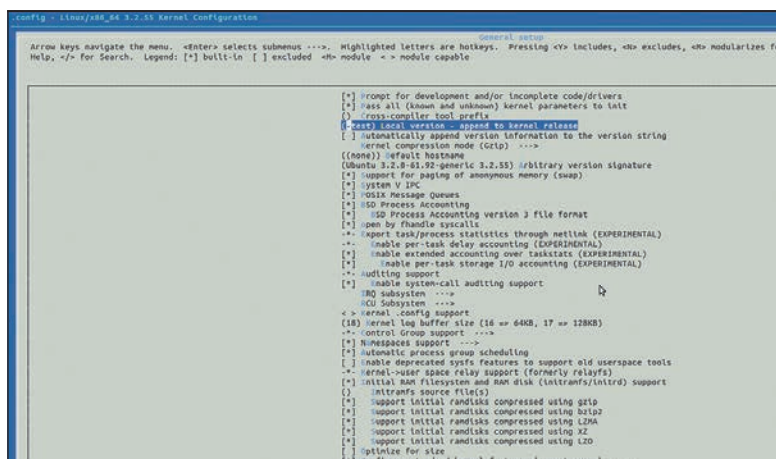
```
cd ../result/_mods
```

```
depmod -b . -v 3.2.55-test
```

Загрузка с новым ядром

Ядро готово. Как его запустить, ничего не сломав? Первое, что приходит в голову — использовать отдельный компьютер, передавая ему во время загрузки образ ядра по сети. Однако в домашних условиях такое решение вряд ли можно назвать практичным. Другой вариант — виртуальная машина. Во многих случаях он бывает неплох, но далеко не всегда, поскольку накладывает ограничения на работу с реальными аппаратными устройствами.

» Конфигуратор ядра Linux.



К счастью, на платформе x86 есть компромиссный путь. Загрузчик *Grub*, применяемый в большинстве современных дистрибутивов Linux, позволяет на одном компьютере установить несколько ОС на разные разделы жесткого диска и выбирать одну из них на этапе загрузки, через меню. Более того, он достаточно гибок, чтобы обеспечить выбор ядра в рамках одной ОС. Если, как в нашем случае, эксперименты касаются только ядра, а корневая файловая система остается неизменной, это как раз то, что надо. К тому же настроить *Grub* на работу в таком режиме достаточно просто, пример можно подсмотреть в Ubuntu: при обновлении ядра Linux новая версия не замещает старую, а добавляется в меню *Grub* как новый вариант загрузки, что дает пользователю возможность использовать любую из предыдущих версий ядра.

Initrd

Едва начав изучать настройки *Grub*, понимаешь, что в процессе загрузки Ubuntu существенную роль играет некий *initrd*. Похоже, с ним тоже придется разбираться.

Initrd (англ. Initial RAM Disk, диск в оперативной памяти для инициализации) — временная файловая система, используемая ядром Linux во время загрузки для монтирования корневого файловой системы (КФС). С ходу ее назначение кажется не очень понятным. Почему бы просто не смонтировать КФС с жесткого диска? Если б ядро содержало все необходимые для этого драйверы, так бы и следовало поступить. Однако Linux поддерживает слишком много ФС и устройств хранения данных, чтобы можно было собрать универсальное ядро разумного размера. Иногда потеря универсальности приемлема. Для остальных случаев служит *initrd*.

Его идея проста: образ ядра оставить небольшим, многочисленными драйверы, о которых шла речь выше, туда не включать, вместо этого собрать их в виде модулей, а при монтировании КФС загружать лишь те из них, что нужны в данной конфигурации. Все бы хорошо, только модули — это обычные файлы, хранящиеся в КФС, но при этом необходимые для ее монтирования. Получается проблема курицы и яйца. Нужно найти способ разорвать этот порочный круг. И вот что для этого придумали разработчики *initrd*. В доступном для *Grub* месте создается RAM-диск (образ *initrd*), содержащий минимальную КФС со всеми необходимыми модулями. *Grub* вместе с образом ядра загружает его в память. Теперь для ядра доступ к модулям заметно упрощается. Ему нужно лишь смонтировать в качестве КФС уже лежащий в памяти образ *initrd*, после чего остается только загрузить нужные модули и смонтировать уже реальную КФС на жестком диске.

Итак, Ubuntu использует *initrd*. Приложив определенные усилия, от этого можно избавиться. Но вряд ли стоит соваться в чужой монастырь со своим уставом. Гораздо проще научиться создавать *initrd* для своих ядер:

```
cd ../result/
mkinitramfs -k -o ./initrd.img-3.2.55-test -r ./_mods/lib/
modules/3.2.55-test/
```

Настройка Grub

Остается настроить загрузчик. В Ubuntu, как и во многих других современных дистрибутивах, используется *Grub 2*. Сначала рассмотрим основные принципы его конфигурирования, а затем, воспользовавшись имеющимися в Ubuntu примерами, добавим новое ядро в качестве еще одного варианта загрузки.

Задача настройки *Grub* решается традиционным для Linux способом — редактированием конфигурационных файлов. В этой версии, в отличие от *Grub 1*, где все параметры сосредоточены в единственном файле — */boot/grub/menu.lst*, она разбивается на две подзадачи: задание общих параметров (например, время ожидания выбора пользователя) и конструирование меню вариантов загрузки.

```
st@dmiltry-M5-7519j:/etc/grub.d
1 | /bin/sh
2 | exec tail -n +3 50
3 | # This file provides an easy way to add custom menu entries. Simply type the
4 | # menu entries you want to add after this comment. Be careful not to change
5 | # the 'exec tail' line above.
6 |
7 | submenu "Test kernels" {
8 | menuentry 'ubuntu-3.2.55-test' --class ubuntu --class gnu-linux --class gnu --class os {
9 |
10 | recordfail
11 | if [ $linux_gfx_mode ]; then
12 | insmod gzio
13 | insmod part_msdos
14 | insmod ext2
15 | set root='(hd0,msdos3)'
16 | search --no-floppy --fs-uuid --setroot b12ec708-4d61-4b5d-98a4-223832141ed1
17 | linux /test/bzImage-3.2.55-test vga=normal root=UUID=75be9ea5-a7d3-41e4-8bde-84666349e58d ro vvt_handoff
18 | initrd /test/Initrd.img-3.2.55-test
19 | }
```

» Отредактированный 40_custom.

За общие параметры отвечает файл */etc/update/grub*. Логика работы с ним следует классической схеме: пользователь вручную вносит изменения, после чего перезагружает ОС, чтобы они вступили в силу. Этот файл можно считать аналогом старого доброго *menu.lst*, но с одной оговоркой: он больше не отвечает за создание меню.

С меню немного сложнее. Оно задается в */boot/grub/grub.cfg*, который, однако, не предназначен для ручного редактирования. Его содержимое генерируется автоматически командой *update-grub* на основе набора *bash*-сценариев в каталоге */etc/grub.d*. Иными словами, правильный путь настройки меню — отредактировать эти сценарии, а затем запустить *update-grub*. Незнание этой особенности может быть источником коварных ошибок. Дело в том, что прямое изменение *grub.cfg* может создать иллюзию успеха, которая продержится до первого вызова *update-grub*, после которого все результаты трудов по редактированию *grub.cfg* будут уничтожены. Это может случиться в самый неожиданный момент — например, в результате установки пакета обновления ядра Linux.

Но довольно теории; пора проверить ее на практике. Создадим подкаталог */boot/test* и скопируем в него наши образы ядра и *initrd*:

```
mkdir -p /boot/test
cp ./bzImage /boot/test/bzImage-3.2.55-test
cp ./initrd.img-3.2.55-test /boot/test/
```

Теперь нужно отредактировать *bash*-сценарии в */etc/grub.d*. Разбираться в их коде не придется. Комментарий в тексте почти пустого *40_custom* недвусмысленно дает понять, что для добавления пунктов в меню загрузки достаточно дописать их в конец этого файла. При вызове *update-grub* они будут просто скопированы в *grub.cfg*. Так и сделаем. Остается разобраться с синтаксисом. Поскольку ничего оригинального не требуется, достаточно будет воспользоваться готовым примером. Возьмем из *grub.cfg* описание пункта меню для текущего ядра Ubuntu и скопируем его в *40_custom*, изменив пути к образам ядра и *initrd*. Там же можно посмотреть пример использования директивы *submenu*. С ее помощью создадим подменю для тестовых ядер.

Ну и последний штрих. Для нормальной работы ядро должно иметь доступ к своим модулям и прошивкам. Оно будет их искать в каталогах */lib/modules/<версия ядра>* и */lib/firmware/<версия ядра>*. Можно создать эти каталоги и скопировать в них нужные файлы или, что еще проще, воспользоваться механизмом символических ссылок.

```
ln -s /home/kernel_test/result/_mods/lib/modules/3.2.55-test/
/lib/modules/3.2.55-test
ln -s /home/kernel_test/result/_mods/lib/firmware/
/lib/firmware/3.2.55-test
```

Все готово, можно проверять. Перезагружаем компьютер, в меню *Grub 2* выбираем подменю *Test kernels*, а в нем единственный (пока что) пункт меню. Может показаться, что ничего не изменилось, загрузка проходит как обычно. Так и должно быть: ведь мы собрали копию рабочего ядра Ubuntu, разница лишь в добавлении *-test* к номеру версии. Вывести ее и, тем самым, удостовериться в корректности настройки загрузчика поможет команда *uname -sr*

»

Отсутствие видимых проблем при загрузке ядра — отличная новость. Но это лишь крайне грубая оценка его работоспособности. Вопреки бытующему мнению, разработчики ядра не столь

аскетичны, а потому имеют на вооружении довольно внушительный, хотя и немного непривычный, арсенал отладочных средств, который будет бегло рассмотрен в оставшейся части статьи.

Отладка ядра

Отладочная печать

Начнем с самого простого — вставки отладочного сообщения. В коде ядра нет доступа к библиотекам, поэтому вызвать привычную `printf` не получится. Вместо этого следует использовать ее «ядерный» аналог — функцию `int printk(const char* fmt, ...)`, имеющую два существенных отличия:

» Перед первым параметром нужно указывать уровень важности сообщения. В коде ядра (`/include/linux/printk.h`) их определено 8: от `KERN_DEBUG` до `KERN_EMERG`.

» Вывод осуществляется в текстовую консоль, ее можно наблюдать на экране при загрузке ядра (если в его параметрах нет `quiet`). Кроме того, все сообщения попадают в кольцевой буфер ядра, содержимое которого показывает команда `dmesg`.

Вот пример вывода строки и целого числа (уровень важности `KERN_DEBUG`):

```
printk(KERN_DEBUG "%s: int value=%d", "MSG", 5);
```

В результате получим:

```
MSG: int value=5
```

Это основное средство отладочной печати общего назначения. Кроме того, есть несколько более специализированных функций и макросов:

» `stack_dump()` Выводит обратную трассировку вызовов функций.

» `BUG()` Заставляет ядро сгенерировать так называемое сообщение «oops», содержащее дампы всех регистров и обратную трассировку вызовов функций.

» `BUG_ON(cond)` Аналогичен `BUG()`, но принимает условие и генерирует «oops» только в случае его истинности.

Модули ядра

Модули — полноправные части ядра Linux, имеющие доступ ко всем его подсистемам. Используя команды `insmod` и `rmmod`, их в любой момент можно подключить к работающему ядру и отключить от него. Таким образом, запустить код ядра, оформленный в виде модуля, почти так же просто, как обычное приложение. Очевидно, при отладке это существенно экономит время. Реализация такого механизма непростая, однако ее детали надежно скрыты в недрах ядра Linux, что позволяет легко научиться писать свои модули.

Создадим отдельный каталог и перейдем в него:

```
mkdir /home/kernel_test/test_mod
```

» Текст простейшего модуля ядра.

```
root@dmity-MS-7519:~/kernel_test/test_mod
1 #include <linux/module.h>
2
3 static int __init test_mod_init(void)
4 {
5     printk(KERN_DEBUG "Init test_mod.\n");
6     return 0;
7 }
8
9 static void __exit test_mod_exit(void)
10 {
11     printk(KERN_DEBUG "Exit test_mod.\n");
12 }
13
14 module_init(test_mod_init);
15 module_exit(test_mod_exit);
16
17 MODULE_AUTHOR("Dmitry Kuznetsov");
18 MODULE_DESCRIPTION("Test module.");
19 MODULE_LICENSE("GPL");
~
```

```
cd /home/kernel_test/test_mod
```

В нем должна быть символическая ссылка на откомпилированное дерево исходного кода ядра:

```
ln -s /home/kernel_test/linux-source-3.2.0 ./build
```

Файл `test_mod.c` с исходным кодом модуля показан на рисунке, его структура очевидна:

» включение `module.h`;

» определение двух функций, которые будут вызваны при подключении и отключении модуля;

» определение имени автора, описания модуля и лицензии с помощью специально предназначенных для этого макросов.

Makefile и того проще — всего из одной строки: `obj-m += test_mod.o`.

Все готово, компилируем:

```
make -C «./build» M=»$PWD»
```

В текущем каталоге должен появиться модуль `test_mod.ko`.

Вот как можно удостовериться, что он действительно работает:

```
insmod ./test_mod.ko
```

```
rmmod test_mod
```

```
dmesg | tail -n 2
```

Отладчики

В прикладном программировании использование отладчиков давно вошло в привычку. Но применить этот опыт для отладки ядра Linux с ходу не получится. Стандартный отладчик `GDB` способен только заглянуть в ядро и вывести из него информацию на консоль. Изменение данных, установка точек останова и пошаговое выполнение остаются недоступны. Для запуска `GDB` в таком режиме нужно указать два параметра:

» `vmlinux` Несжатый образ ядра, создаваемый при его сборке.

» `/proc/kcore` Образ памяти компьютера (представлен в виде файла в `procfs`).

Попробуем:

```
cd /home/kernel_test/linux-source-3.2.0
```

```
gdb ./vmlinux/proc/kcore
```

После ряда сообщений появляется приглашение командной строки `GDB` — (`gdb`). Отладчик готов к работе. Для его проверки выведем значения какой-нибудь структуры данных ядра, корректность которых легко оценить. Хороший пример — `static struct file_system_type *file_systems (/fs/filesystems.c, строка 32)`. Это указатель на первый элемент списка структур, представляющих драйверы файловых систем. У каждого элемента есть поле `name`, в котором хранится имя соответствующей файловой системы, и поле `next` — указатель на следующий элемент (`/include/linux/fs.h, строка 1888`). Посмотрим, что в первых двух:

```
(gdb) p file_systems->name
```

```
$2 = 0xfffffff81a25f03 «sysfs»
```

```
(gdb) p file_systems->next->name
```

```
$3 = 0xfffffff81a07a20 «rootfs»
```

Получились вполне правдоподобные имена файловых систем.

И еще одна интересная возможность — команда `disassemble`. Она позволяет получить ассемблерный код любой функции. Например, для `printk` команда выйдет так:

```
(gdb) disassemble printk
```

Ее вывод слишком велик, чтобы быть приведенным здесь.

Для более полноценной работы отладчика в ядре должна быть реализована соответствующая функциональность. В официальной версии ее нет, однако широко известен сторонний проект,

устраняющий этот недостаток: *kgdb*. Он распространяется в виде заплаток-патчей к исходному коду ядра и позволяет с помощью обычного *GDB* отлаживать ядро Linux с удаленного терминала, подключенного к COM-порту нуль-модемным кабелем. Поскольку конфигурация из двух компьютеров малоприменима в домашних условиях, ограничимся лишь упоминанием о нем.

Встроенные средства. Трассировщики

Разумеется, на отладчиках свет клином не сошелся. Ядро Linux обладает собственной развитой системой отладочных средств, работающих без дополнительных инструментов. Особенности жанра журнальной статьи не позволяют вдоволь позанудствовать, подробно описывая многочисленные параметры сборки ядра, включающие вывод отладочных сообщений для различных драйверов и подсистем. По той же причине оставим в стороне большую часть содержимого псевдофайловых систем *sysfs*, *procfs* и *debugfs*, традиционно монтируемых в каталоги */sys*, */proc* и */sys/kernel/debug* и дающих возможность с помощью обычных средств ОС для работы с файлами получить огромный объем информации о работе ядра. Остановимся лишь на трассировщиках, обойти вниманием которые при сколь-нибудь полном обсуждении вопросов отладки было бы непростительным упущением.

Этот весьма полезный отладочный механизм встроен в ядро Linux (при сборке включается в разделе Kernel hacking > Tracers), а интерфейс управления им доступен через каталог *tracing* в *debugfs*. На первый взгляд он может показаться довольно сложным: **ls /tracing** выводит список из нескольких десятков элементов. Однако такое впечатление ошибочно. Понять основные принципы обращения с трассировщиками и начать ими пользоваться довольно просто. Более глубоких знаний можно набраться в ходе дальнейшей работы, экспериментируя и время от времени заглядывая в документацию в дереве исходного кода ядра (**Documentation/trace**). Итак, необходимый минимум:

» Для удобства перейдем в упомянутый выше каталог:

```
cd /sys/kernel/debug/tracing
```

» Ядро поддерживает несколько видов трассировщиков. Список доступных в данной конфигурации можно прочитать из файла *./available_tracers*:

```
cat ./available_tracers
```

Результат может быть таким (зависит от конфигурации ядра):

```
blk function_graph mmiotrace wakeup_rt wakeup function nop
```

Для демонстрации возьмем простой трассировщик вызовов функций *function*, который просто протоколирует вызовы функций в коде ядра (*function_graph* делает то же самое, но выводит информацию в более удобочитаемом виде).

» Файл *./current_tracer* содержит активный в данный момент трассировщик. Изначально все они отключены, поэтому

```
cat ./current_tracer
```

выводит "nop". Это пустая заглушка, трассировщик, который ничего не делает. Чтобы включить нечто более ценное, достаточно вписать имя нужного трассировщика в этот файл. Например, *function*:

```
echo function > ./current_tracer
```

Теперь результаты трассировки записываются в специальный кольцевой буфер ядра. Его не нужно путать с аналогичным буфером отладочных сообщений от *printk*. Это две совершенно разные вещи, хотя и похожие.

» Чтобы вывести результат трассировки (содержимое кольцевого буфера), нужно прочитать файл *./trace*:

```
cat ./trace
```

Вот одна из строчек этого вывода:

```
Xorg-1309 [000] 13980.191030: os_get_current_thread
<-_nv014907rm
```

Понимать ее следует так: «На 13980.191030 секунде с момента старта системы из функции *_nv014907rm* произошел вызов

```
ltry-M5-7519: /sys/kernel/debug/tracing
cat-6154 [003] 19576.723971: find_get_page <-filenop_fault
cat-6154 [003] 19576.723971: cond_resched <-filenop_fault
cat-6154 [003] 19576.723971: raw_spin_lock <-do_fault
cat-6154 [003] 19576.723972: add_mm_counter_fast <-do_fault
cat-6154 [003] 19576.723972: page_add_file_rmap <-do_fault
cat-6154 [003] 19576.723972: unlock_page <-do_fault
cat-6154 [003] 19576.723972: page_waitqueue <-unlock_page
cat-6154 [003] 19576.723972: wake_up_bit <-unlock_page
cat-6154 [003] 19576.723973: up_read <-do_page_fault
cat-6154 [003] 19576.723974: sys_newstat <-system_call_fastpath
cat-6154 [003] 19576.723975: vfs_fstat <-sys_newstat
cat-6154 [003] 19576.723975: fget_raw_light <-vfs_fstat
cat-6154 [003] 19576.723975: vfs_getattr <-vfs_fstat
cat-6154 [003] 19576.723975: security_inode_getattr <-vfs_getattr
cat-6154 [003] 19576.723975: apparmor_inode_getattr <-security_inode_getattr
cat-6154 [003] 19576.723976: common_perm <-apparmor_inode_getattr
cat-6154 [003] 19576.723976: generic_fillattr <-vfs_getattr
cat-6154 [003] 19576.723976: cp_new_stat <-sys_newstat
cat-6154 [003] 19576.723976: cond_resched <-cp_new_stat
cat-6154 [003] 19576.723977: sys_open <-system_call_fastpath
cat-6154 [003] 19576.723977: do_sys_open <-sys_open
cat-6154 [003] 19576.723978: getname <-do_sys_open
cat-6154 [003] 19576.723978: getname_flags <-getname
cat-6154 [003] 19576.723978: kmem_cache_alloc <-getname_flags
cat-6154 [003] 19576.723979: cond_resched <-kmem_cache_alloc
cat-6154 [003] 19576.723979: do_getname <-getname_flags
cat-6154 [003] 19576.723979: strncpy_from_user <-do_getname
cat-6154 [003] 19576.723979: strncpy_from_user <-strncpy_from_user
cat-6154 [003] 19576.723979: cond_resched <-strncpy_from_user
cat-6154 [003] 19576.723979: alloc_fd <-do_sys_open
cat-6154 [003] 19576.723979: raw_spin_lock <-alloc_fd
cat-6154 [003] 19576.723980: expand_files <-alloc_fd
cat-6154 [003] 19576.723980: do_filp_open <-do_sys_open
cat-6154 [003] 19576.723980: path_openat <-do_filp_open
cat-6154 [003] 19576.723980: get_empty_filp <-path_openat
cat-6154 [003] 19576.723980: kmem_cache_alloc <-get_empty_filp
cat-6154 [003] 19576.723981: cond_resched <-kmem_cache_alloc
cat-6154 [003] 19576.723981: security_file_alloc <-get_empty_filp
cat-6154 [003] 19576.723981: apparmor_file_alloc_security <-security_file_alloc
cat-6154 [003] 19576.723981: kmem_cache_alloc_trace <-apparmor_file_alloc_security
cat-6154 [003] 19576.723982: cond_resched <-kmem_cache_alloc_trace
cat-6154 [003] 19576.723982: path_init <-path_openat
cat-6154 [003] 19576.723982: vfsmount_lock_local_lock <-path_init
cat-6154 [003] 19576.723982: link_path_walk <-path_openat
cat-6154 [003] 19576.723983: inode_permission <-link_path_walk
cat-6154 [003] 19576.723983: generic_permission <-inode_permission
cat-6154 [003] 19576.723984: acl_permission_check <-generic_permission
cat-6154 [003] 19576.723984: security_inode_permission <-inode_permission
cat-6154 [003] 19576.723984: cap_inode_permission <-security_inode_permission
cat-6154 [003] 19576.723984: inode_permission <-link_path_walk
```

функции *os_get_current_thread*. Это случилось в процессе Xorg с идентификатором 1309, на процессоре 000».

Объем информации впечатляет, но, очевидно, слишком велик, чтобы в нем было легко ориентироваться. Для решения этой проблемы предусмотрен ряд методов:

» Очистить кольцевой буфер трассировки и начать работу с чистого листа. Для этого достаточно переключить текущий трассировщик через *./current_tracer* или выключить трассировку, записав в файл *./tracing_enabled* 0 и 1 соответственно.

» Добавлять свои сообщения в результаты трассировки. Для этого текст сообщения нужно записать в файл *./trace_marker*. Вот простейшей пример:

```
echo hello > ./trace_marker
```

Поскольку вся суть этого трюка сводится к обычной записи в файл, его можно выполнять не только из командной строки, как показано выше, но и из своей программы. Например, в языке C можно воспользоваться функциями *open*, *close*, *read*, *write*. Это что касается пользовательского режима. В режиме ядра все и того проще: достаточно вызвать *trace_printk*. Ее использование аналогично уже знакомой *printk*.

» Приостанавливать и возобновлять трассировку, записывая в файл *./tracing_on* 0 или 1 соответственно. В коде ядра аналогичная функциональность реализуется вызовами функций *tracing_off()* и *tracing_on()*.

Умелое комбинирование этих методов превращает трассировщики в очень гибкий и удобный инструмент, легко настраиваемый на применение в нужное время и в нужном месте. Вот, например, как посмотреть, что творится в ядре при выполнении команды **ls**:

```
echo 0 > ./tracing_enabled
```

```
echo 1 > ./tracing_enabled
```

```
echo 0 > ./tracing_on
```

```
echo 1 > ./tracing_on; ls; echo 0 > ./tracing_on
```

```
echo cat ./trace
```

Заключение

Как часто оказывается, черт не так страшен, как его малюют. А ядро ОС — всего лишь обычная программа для ЭВМ. Кто-то возразит, что это не совсем так, что в теории ОС есть своя специфика. Безусловно, с этим трудно спорить. Но в какой предметной области ее нет? Если есть желание действовать, всегда можно взять и разобраться. Надеюсь, в этой статье удалось показать несколько первых шагов этого не всегда легкого, но непременно интересного пути. Удачного хакинга! **LXF**

» Трассировщик function в действии.

ПЛИС. Игрушки для взрослых

Михаил Остапкевич рассказывает о чудесах программируемой электроники, доступных почти каждому, а Евгений Балдин проверяет каждое его слово.



Наш эксперт

Михаил Остапкевич
Романтик, очарованный компьютерами и создаваемыми в них идеальными мирами.

Но что же это значит — ПЛИС? ПЛИС — это акроним от Программируемая Логическая Интегральная Схема, что, в свою очередь, является аналогом Programmable Logic Device [Устройство с программируемой логикой] или PLD. Такая схема позволяет оперативно перенастраивать себя на выполнение той или иной функции. Это её основное отличие от большинства других микросхем, функция которых жёстко задаётся при их изготовлении и далее меняться не может. То есть, как правило, контроллеру интерфейса не светит превращение в микропроцессор, если они не допускают реализации на ПЛИС.

Зачем нужны ПЛИС?

Современное состояние дел сейчас таково, что заказ и изготовление одной микросхемы или ста тысяч микросхем фактически не отличаются ни по деньгам, ни по временным затратам. Разница, конечно, есть, причем существенная: во втором случае одна микросхема будет стоить во много тысяч раз меньше, поскольку разработка микросхемы и подготовка её к печати стоит дорого, а репликация — дешево. Экономически совершенно не целесообразно заказывать специальную микросхему для уникального проекта — или не очень уникального, но редко встречающегося в природе и интересного только немногочисленным профессиональным исследователям или энтузиастам. Самое интересное начинается, когда в пришедшей свеженькой, прямо «из печки» партии микросхем обнаруживается ошибка, и производство нужно начинать заново.

И здесь приходят на помощь ПЛИС, так как большой тираж делает их относительно недорогим средством для реализации разного рода нестандартных схем. Воспользовавшись настраиваемостью ПЛИС, можно «прошить» в неё нужную функцию. Большим достоинством гибкости перенастройки является то, что в случае обнаружения ошибки в реализации функции для её исправления потребуется всего лишь перепрограммировать ПЛИС, а это значительно проще заказа новой партии микросхем на фабрике.

Изначально ПЛИС использовали в схемах управления. Теперь, когда число ячеек в них огромно, а тактовые частоты постепенно

приближаются к тактовым частотам передовых микропроцессоров, их стали использовать и для очень высокопроизводительной обработки информации.

Перечень разработок на ПЛИС также огромен. В основном они скрыты от общества в недрах многочисленных коммерческих компаний, но есть и свободные наработки. Представление о спектре доступных к использованию открытых проектов даёт портал opencores.org. Там есть разнообразные арифметические и криптографические ядра, полноценные процессоры, контроллеры памяти и сетевых интерфейсов, видеокарты с генерацией 3D-изображений и многое другое.

Как устроены ПЛИС?

Существуют различные виды ПЛИС. Здесь мы подробно остановимся на Field Programmable Gate Array (FPGA) или Программируемые Пользователем Вентильных Матрицах (ППВМ).

ППВМ состоит из большого числа блоков нескольких типов. Больше всего в наличии логических [gates] и коммуникационных [routers] блоков. Логические блоки можно запрограммировать на реализацию той или иной логической функции. В них есть память для хранения таблицы истинности и хранения результата функции (если нужен синхронный режим работы). Число входных параметров различно для разных моделей ППВМ.

Если требуется нечто простенькое, как на приведенном рисунке, то проблем нет; но что же делать, если требуемая функция имеет больше входов или очень сложна? В этом случае нужно задействовать больше логических блоков.

Однако тогда встает проблема соединения блоков. Причём речь идёт не о жёстко заданном типе соединения, а о достаточно произвольных связях. Передачей сигналов между блоками всех остальных типов заняты коммуникационные блоки. В каждом из них есть программируемый коммутатор соединений и, конечно, память для него, хранящая конфигурацию коммутатора. В итоге коммуникационный блок можно настроить на реализацию локальных связей между соседними логическими блоками, или, наоборот, на создание очень длинной (глобальной) связи между очень удалёнными блоками.

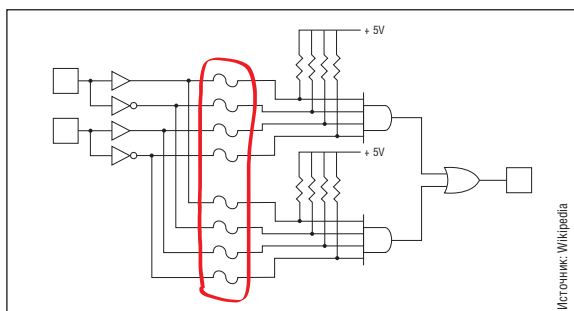
ППВМ существует не сама по себе, а является частью некоторой цифровой схемы. Она может управлять другими устройствами, передавать им данные и, наоборот, получать команды и данные извне. Для реализации таких внешних связей в ППВМ предусмотрены блоки ввода/вывода. С одной стороны, они могут соединяться с другими блоками на ППВМ. С другой стороны, они подключаются к выводам микросхемы, которые обеспечивают доступ во внешний мир. Ещё один аспект гибкости ППВМ заключается в том, что выводы можно настроить на различные электрические интерфейсы с различным кодированием двоичного сигнала: TTL, CMOS, LVCMOS, PCI и т.д.

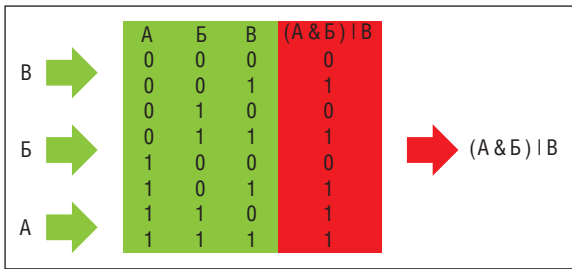


Наш эксперт

Евгений Балдин
Физик, который действительно знает, что такое нехватка вычислительных ресурсов.

» Упрощённая ПЛИС. Программируемые элементы здесь представлены в виде переключателей внутри красного контура. Логические сигналы с двух входов, как и их инвертированные версии, собираются в две логические схемы И, результаты которых объединяются по схеме ИЛИ.



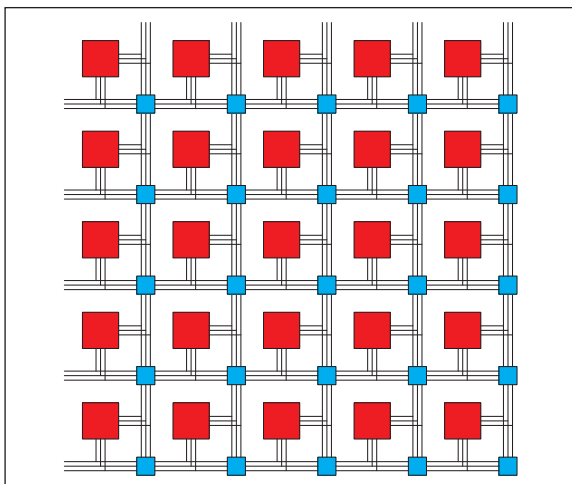


▶ Пример логического блока для трёх входов и одного выхода.

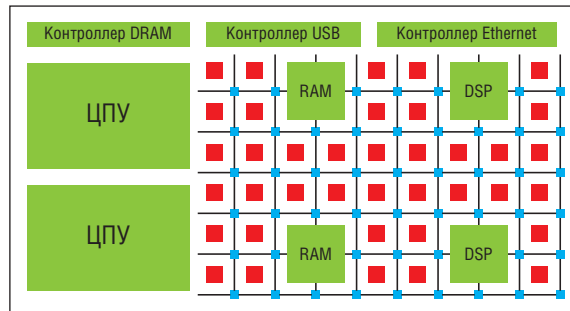
Остальные виды блоков, присутствующих внутри ППВМ в меньших количествах, решают более специфичные задачи. Некоторые из них имеют значительно более сложную внутреннюю организацию. На ППВМ обычно имеется несколько блоков управления синхросигналом. Они позволяют при фиксированной частоте кварцевого резонатора на плате передавать другим блокам ППВМ синхросигнал на разных частотах. Блоки памяти (совокупности ячеек статического ОЗУ) позволяют создавать регистровые файлы, буферы. Обычно на кристалл добавляют несколько умножителей. Они производят умножение существенно эффективнее, чем умножители, построенные из логических блоков. Современные ППВМ могут включать и ещё более сложные блоки, такие как сигнальные процессоры, контроллеры USB, DDR3, Ethernet и даже полноценные ядра процессоров.

Платы разработчика

Как и в случае с Arduino, программисту или разработчику программно-аппаратного приложения не всегда удобно самому разрабатывать и паять оборудование на основе микросхем. В случае с ППВМ это еще сложнее, ведь крошечная микросхема может иметь более тысячи выводов. Для многих штатных случаев, когда нет специфических требований, вполне подходят универсальные платы разработчика. В простейшем случае они содержат саму ППВМ, интерфейс для её программирования, схему питания и удобные разъемы для подключения периферии. Наиболее продвинутые платы включают память разных сортов, светодиоды и дисплей для индикации, кнопки и переключатели и разнообразные интерфейсы ввода/вывода. Обычной практикой также является выпуск плат расширений, подключаемых к платам разработчика, которые призваны реализовать дополнительную функциональность: например, ввод аналоговых данных или поддержку сенсорного экрана.



▶ Пример простого однородного ППВМ, состоящего только из логических (красные квадратики) и коммуникационных (синие квадратики) блоков.



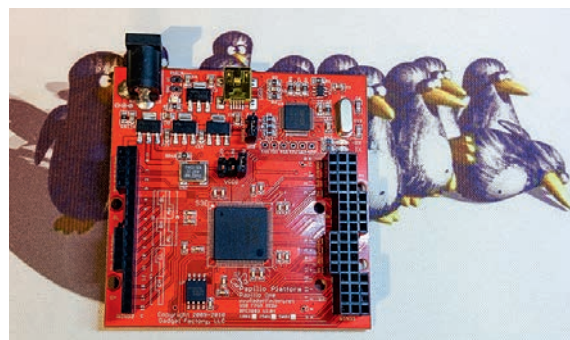
▶ Сейчас ППВМ состоит не только из логических и коммуникационных блоков, но и содержит «стандартные» непрограммируемые элементы (зелёные прямоугольники).

Плата разработчика Papilio One 500K

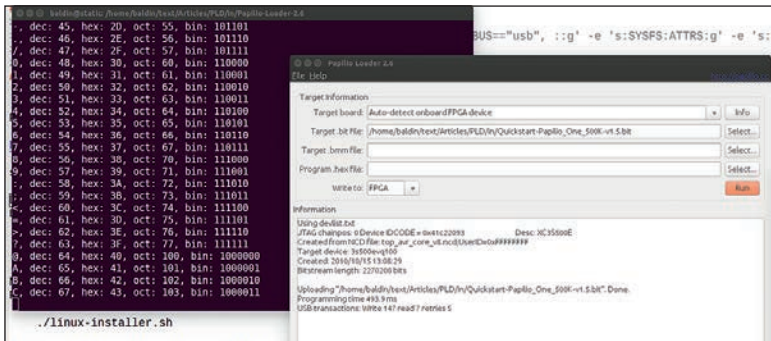
У основных производителей ППВМ, Xilinx и Altera, имеется несколько совсем недорогих серий моделей плат разработчика. У Xilinx наиболее известные из них — это Basys и Papilio из семейства Spartan 3E (papilio.cc). Здесь мы рассмотрим возможности серии Papilio [бабочка-парусник], точнее, одной из ее моделей — Papilio One 500K. Серия разработана в Gadget Factory Джеком Гассеттом [Jack Gasset]. Свою разработку Джек выпустил под лицензией Creative Commons — Attribution-NonCommercial-ShareAlike 3.0 Unported License, ознакомиться с которой можно на сайте Creative Commons, creativecommons.ru/licenses. Данная лицензия позволяет менять и распространять исходный код при условии сохранения лицензии на производные, но не разрешает применять созданное с использованием этих исходников железо и его модификации для коммерческих целей. Иными словами, лицензия не является свободной в полном смысле этого слова, но для целей обучения и для реализации «домашних проектов» её хватает.

Плата Papilio имеет размер 7×7×1 см и цену около 2600 рублей при заказе через eBay (на сайте разработчика — \$88). Она построена на базе ППВМ Xilinx Spartan 3E XC3S500E. По современным меркам эта ППВМ имеет скромные ресурсы: 500K системных вентилях, 10 476 логических блоков, 73 Кбита распределенных ячеек памяти, 360 Кбит в блоках ОЗУ, 20 18-разрядных умножителей, 4 блока управления синхросигналом и 232 контакта ввода/вывода. Скорость передачи по одному каналу до 633 Мбит/с, а частота синхросигнала — от 5 до 300 МГц. Кроме ППВМ, на плате есть поддержка интерфейса USB с разъемом Mini USB. Он используется как для программирования, так и для передачи данных между платой и компьютером. Кроме того, Mini USB можно использовать и для питания.

При программировании устройства конфигурация может записываться либо непосредственно в ППВМ, либо в имеющуюся flash-память объемом 4 Мбит. Во втором случае конфигурация сохраняется и при выключенном питании. Свободную часть flash-памяти можно использовать для хранения данных. Для подключения периферийных каналов ввода/вывода имеется 48 контактов ввода/вывода. Gadget Factory разработал различные платы-расширения: блок кнопок, вывод VGA — на монитор, адаптер SD-карт, аудиовывод, драйвер шаговых двигателей и т. д., которые придётся приобретать отдельно.



▶ Это героиня нашего рассказа: плата разработчика Papilio One 500K. Невысокая стоимость платы делает её доступной для обучения и небольших проектов.



➤ Загрузка программы-примера в плату Papilio. В терминале виден результат деятельности программы — выводится ASCII-таблица символов.

Papilio Loader

Прежде чем начинать что-то делать, нужно почитать документацию. В нашем случае полезно проглядеть Papilio Quick Start Guide на сайте papilio.cc. Для загрузки программы в виде bit-файлов (как их готовить, будет рассказано чуть ниже) в ППВМ в среде GNU/Linux используется *Papilio Loader*. Этот программный пакет содержит Java-GUI (*papilio-loader-gui*) и консольную утилиту (*papilio-prog*) для загрузки программ в плату. Скачайте эту программу с forum.gadgetfactory.net (есть ссылка в Papilio Quick Start Guide). Для установки нужно выполнить следующие действия:

```
sudo apt-get install libftdi1:i386 libftdi-dev default-jdk
tar -xzf Papilio-Loader-2.6.tar.gz
cd Papilio-Loader-2.6/
sudo ./linux-installer.sh
```

Обратите внимание, что нужна именно 32-битная версия библиотеки *libftdi*. По умолчанию программа устанавливается в `/opt/GadgetFactory/papilio-loader/`.

Теперь не мешает проверить работоспособность платы. Для этого нужно скачать тестовую программку *HelloWorld* (на неё есть ссылка в Papilio Quick Start Guide) и запустить *Papilio Loader*:

```
wget http://papilio.cc/sketches/Quickstart-Papilio_One_500K-v1.5.bit
papilio-loader-gui
```

В программе следует выбрать скачанный bit-файл и записать его в ПЛИС.

Предлагаемая для тестирования программа выполняет несколько действий, но нас интересует то, что она посылает в бесконечном цикле таблицу ASCII-символов в виртуальный последовательный порт через USB. Поглядеть на эту таблицу можно с помощью команды `screen`:

```
sudo screen /dev/ttyUSB1 9600
```

Налюбовавшись, можно выйти из *screen* с помощью комбинации клавиш `Ctrl + a`.

Инструментарий разработчика

Существует несколько вариантов разработки на Papilio. Можно, например, запрограммировать ППВМ как программный процессор с архитектурой AVR8, как это делается в Arduino Uno, и использовать модифицированную среду Arduino от Gadget Factory. Или можно превратить модуль Spartan 3E в более быстрый, чем AVR8, RISC-процессор ZPU (opencores.org) с тактовой частотой 95 МГц и получить производительность уже на уровне Arduino Due.

На этом уроке мы сосредоточимся на средствах разработки от самой фирмы Xilinx. У Xilinx два основных пакета — *ISE* (базовый) и *Vivado* (более продвинутый, для более современных ППВМ). Фирма бесплатно распространяет *ISE WebPACK Design Software* (www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.htm). Им мы и воспользуемся. На сайте www.xilinx.com заходим в меню Products и тыкаем в раздел ISE Design Suite. Напоминаю, что нам нужен *ISE WebPACK Design Software*, который мы и попытаемся скачать.

Чтобы скачивать это программное обеспечение, следует зарегистрироваться на сайте Xilinx. При этом необходимо указать

«корпоративную электронную почту», место работы, адрес работы (в адресных строках допускаются только буквы и цифры — форма не переваривает запятых), телефон на работе, город и страну. В общем, суровый «энтерпрайз» в полный рост.

Пакет инсталлятора *ISE* (текущая версия — 14.7) имеет весьма внушительный объём в 6 гигабайт, поэтому, запустив закачку, вы успеете пообедать.

Для установки необходимы права root. Если не хочется делать это в самой системе, можно посадить программу в песочницу, например с помощью механизмов `docker.io`. Подробности можно найти в блоге Христиана Янна [Christian Jann] <http://www.jann.cc>. Там же есть статьи, посвящённые установке и работе с ISE.

```
tar -xzf Xilinx_ISE_DS_Lin_14.7_1015_1.tar
cd Xilinx_ISE_DS_Lin_14.7_1015_1
sudo ./xsetup
```

При установке выберите *ISE WebPack*. После этого не нужно добавлять установку драйвера USB-кабеля. Всё равно ничего не получится (энтерпрайз, такой энтерпрайз...). Получившиеся скомпилированные bit-файлы будем загружать с помощью *Papilio Loader*. Всё устанавливается по умолчанию в `/opt/Xilinx` и требует для этого 18 Гб.

Затем нужно получить бесплатную лицензию. Для этого запустим ISE (можно сделать соответствующий скрипт):

```
source /opt/Xilinx/14.7/ISE_DS/settings64.sh
ise
```

При первом запуске всплывает сообщение: “A license was not found for the ISE feature. [Лицензия не найдена]” Соглашаемся с ним и выбираем WebPACK license, затем сохраняем информацию на диске в виде файла *Xilinx_Connect_Later.html*, открываем этот файл в браузере и заходим на сайт Xilinx, где опять выбираем WebPACK license. После некоторого количества форм и сообщений файл с лицензией *Xilinx.lis* высылают на e-mail. Этот файл импортируется в *ISE*. Уф-фф, теперь можно что-нибудь поделать.

Помигаем светодиодам

Для затравки проведём небольшую «лабораторную работу» — её целью будет помигать присоединёнными к плате внешним светодиодам (который нужно приобрести отдельно).

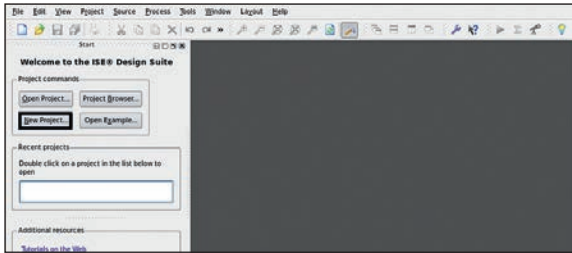
Для описания цифровой логики воспользуемся языком VHDL — точнее, малым его подмножеством. В VHDL существует два основных способа описывать логику работы цифровой схемы. Один — структурный. Он заключается в составлении композиции блоков заранее определённых видов. При добавлении блока в композицию все его входы и выходы должны быть соединены с какими-либо сигналами, например, с выходами или входами других блоков композиции.

Но для построения нашей модели мы употребим второй способ: опишем поведение цифровой схемы. Тут нам пригодится языковая конструкция `process`. Она запускает операторы внутри себя каждый раз, когда меняется значение какого-либо из сигналов, перечисленных в заголовке блока.

Для начала подготовим себе проект, как показано в пошаговой инструкции на стр. 83 вверху. В сгенерированном исходном файле уже есть пустое описание модуля (блок entity) и пустая реализация (блок architecture). Вставим в него следующий текст:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
--use IEEE.STD_LOGIC_ARITH.ALL;
--use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity main is
Port (
led0 : out STD_LOGIC;
clk : in STD_LOGIC
);
end main;
```

Подготовим себе проект



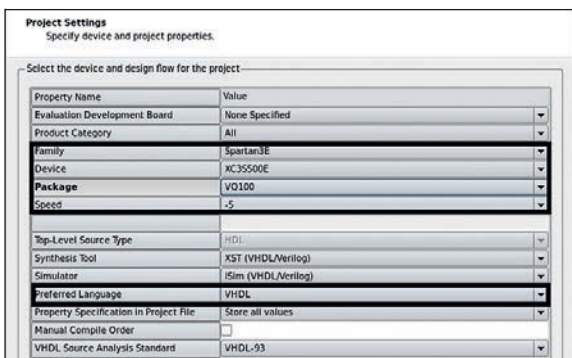
1 Запускаемся

Запускаем WebPack и приступаем к созданию нового проекта, нажав кнопку **New Project**. Перед нами появится диалоговое окно создания проекта.



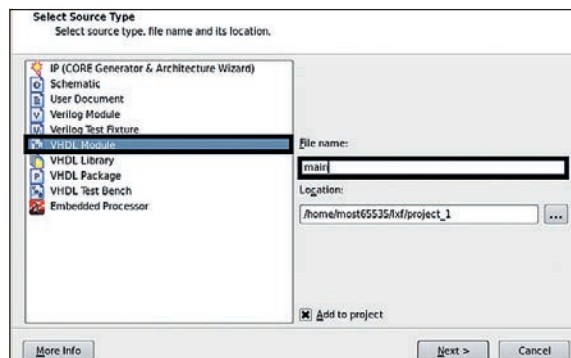
2 Создаём проект

В первой вкладке появившегося окна задаем имя проекта и директорию, где он будет располагаться. Переходим к следующей вкладке нажатием клавиши **Next**.



3 Готово!

Во второй вкладке выбираем настройки семейства ППВМ (Spartan 3E), модели (XC3S500E), типа корпуса (CP132), скорости (-5) и предпочтительного языка (VHDL). Нажимаем **Next** на этой вкладке и **Finish** на следующей. Проект создан!



4 Добавим свою логику

Создаем новый исходный текст с добавлением к проекту, выбрав в меню **Project** пункт **New Source**. Выбираем **VHDL Module** в качестве типа исходного файла. Задаем имя файла **main**. Два раза нажимаем **Next** и один раз **Finish**.

```
architecture Behavioral of main is
    signal counter : STD_LOGIC_VECTOR(24 downto 0) := (others => '0');
    signal state1 : STD_LOGIC := '0';
begin
    process(clk, counter)
    begin
        if (clk'event and clk = '1') then
            counter <= counter - 1;
            if counter = 0 then
                counter <= "111111111111111111111111";
                state1 <= not state1;
            end if;
        end if;
    end process;
    led0 <= state1;
end Behavioral;
```

В программе описывается входной сигнал от кварцевого генератора (32 МГц). Здесь мы делим его, чтобы получить период мигания в 1 сек и выходной сигнал для светодиода, который будет мигать.

Чтобы установить связь между абстрактными входными и выходными сигналами внутри описания на VHDL и реальными контактами микросхемы, подключенными к тому или иному устройству, применяется файл ограничений. В файле ограничений происходит назначение таких контактов по их идентификатору. Также там указывается, какой электрический интерфейс необходимо использовать.

В случае нашей первой модели нам необходимо описать входной синхросигнал и выходной сигнал, подаваемый на светодиод. Вход для синхросигнала мы выбирать не можем. Кварцевый резонатор на плате уже впаиван на фабрике, и его тактовая частота не меняется. Для *Papilio* это 32 МГц, что требует от нас указать период 31,25 нс. А вот для подключения светодиода мы можем выбирать. Выберем, например, контакт P17. В итоге получается такой файл:

```
NET «clk» LOC = «P89» | IOSTANDARD =
LVCMOS25 | PERIOD = 31.25ns ;
NET led0 LOC = «P17» | IOSTANDARD=LVTTL;
```

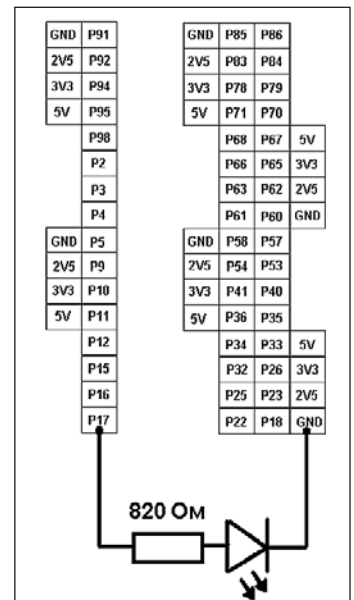
Сохраним его как **c1.ucf** и добавим в проект (в диалоге добавления нового исходника выбираем тип **Implementation Constraint File**).

Теперь подключим светодиод и резистор, как показано на рисунке.

Подключаем плату к USB хост-машины, генерируем файл настройки ППВМ (bit-файл), с помощью *Papilio Loader* или командной утилиты загружаем bit-файл в плату... Всё. По идее, светодиод должен теперь замигать.

В следующей статье серии будет рассказано о том, как организовать связь между *Papilio* и Linux-машиной с помощью передачи данных через виртуальный последовательный порт поверх USB. **LXF**

➤ Подключение светодиода и резистора.



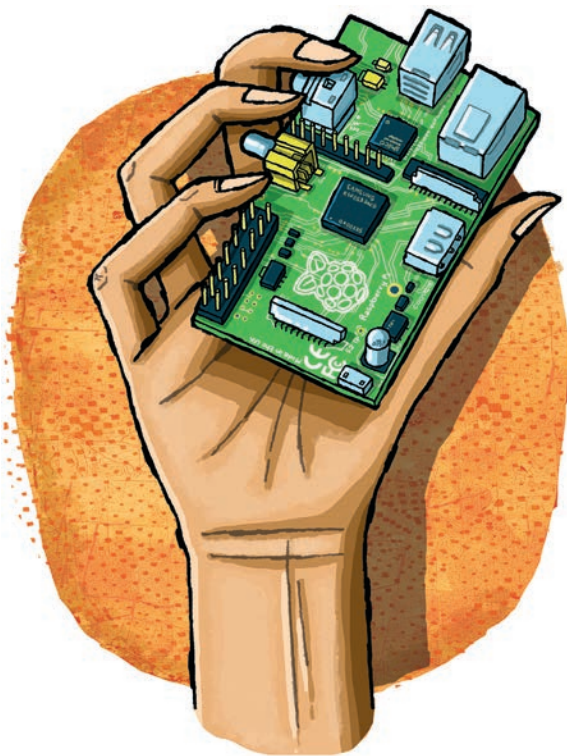
Python: Строим кластер из Pi

Джонни Бидвелл соединяет несколько Pi в кластер, любовно названный Bramble, и направляет его мощь на взлом паролей.



Наш эксперт

Джонни Бидвелл любит Linux, Python, криптографию, Raspberry Pi и лето. Он от души рекомендует прочесть «Криптономикон» поздно ночью, сидя в деревянной избушке в глухом лесу.



Когда-то было принято объединять компьютеры (не обязательно сверхбыстродействующие) в сеть, чтобы нацелить их мощности на одну задачу или на группу связанных задач. К примерам такого объединения относятся фермы для рендеринга и моделирования, когда задача легко распределяется между узлами. Подобные кластеры отчасти вышли из моды — специализированное или промышленное «железо» дает гораздо лучшее соотношение «результативность/мощность», а в многоядерном процессоре обмен данными происходит быстрее, чем по сетевому кабелю. Но не дадим трендам преградить путь старым добрым самоделкинским.

Колючие вопросы

Общее название для таких сетей — кластер Beowulf. Это имя персонажа классической одноименной англосаксонской поэмы, который «тридцать ратников переборол одной рукою» [цит. по переводу на <http://norse.ulver.com/src/other/beowulf/beowulf.html>, — прим. пер.]. Поскольку вычислительными устройствами служат только Raspberry Pi, то сообщество RPi предложило новое название кластера: Bramble [англ. ежевика; Raspberry — малина]. Хотя большинство настольных компьютеров примерно на порядок (в зависимости от задачи) быстрее Pi, заниматься распределенными

вычислениями на Python на кластере из Pi недорого и приятно. Для этой демонстрации мы воспользовались двумя Pi с питанием от хаба PiTogoPi (с отдельным портом для каждой Pi), подключенным к пятипортовому коммутатору на 100 Мбит/с (его можно купить за £5 на eBay). К хабу можно подключать до пяти Pi, поэтому когда-нибудь наш кустарник разрастется.

На нашем уроке мы будем атаковать методом перебора два хорошо известных алгоритма свертки: MD5 и SHA-1. Точнее, мы попробуем найти исходный пароль (образ) по известному значению свертки-хэша, которое мы вычислим для (довольно простой) парольной фразы. Хотя оба этих алгоритма уязвимы к атакам методом коллизии (цель которых — найти любую пару входных значений, дающих одну и ту же свертку), атаки с поиском исходного пароля гораздо труднее. Такие атаки для MD5 существуют, но требуют чрезмерных затрат времени. Для SHA-1 на момент написания статьи такие атаки не известны. Вредоносная программа Flame в 2012 году выявила уязвимость к атакам методом коллизии для MD5 для подделки сертификатов безопасности Microsoft.

Напишем малость кода

Мы часто слышим об атаках на базы данных, которые приводят к утечкам свертков паролей. Хотя использование «соли [добавление случайных данных]» помогает защититься от некоторых атак, пароли остаются уязвимыми к атакам по словарю, поэтому возможность такого взлома следует воспринимать серьезно. Мы в своем коде не будем реализовывать атаки по словарю, сосредоточившись вместо этого на элементарных комбинациях символов. Словари можете реализовать сами, загрузив готовые списки слов или создав собственные с помощью таких утилит, как *Crunch*. Потом с помощью простого кода можно разбить этот список и поделить задание между исполнителями.

Все функции криптографической свертки доступны в стандартной установке Python в библиотеке *hashlib*. Например, для получения свертки MD5 пароля "password1" в Python воспользуемся следующим кодом:

```
>>> import hashlib
>>> h = hashlib.md5('password1')
>>> h.hexdigest()
'7c6a180b36896a0a8c02787eeafb0e4c'
```

Заменив md5 на sha1, мы получим соответствующую свертку для SHA-1:

```
'db6a4668837a519ccef4616751e41674807d6a8a'
```

На нашем уроке мы попробуем найти пароль для следующей свертки MD5:

```
target_hash_md5 = '392b2bd9f1571ff02e7dc21adfb2f0b0'
```

Предположим, нас интересуют только пароли, состоящие из заглавных и строчных букв и чисел от 0 до 9. Это легко сделать, объединив соответствующие диапазоны кодов ASCII:

```
ascii = range(48,59) + range(65,91) + range(97,123)
```

```
chars = [chr(j) for j in ascii]
```

Для генерации всех возможных комбинаций указанной здесь длины можно воспользоваться модулем *itertools*. Это позволит нам не писать утомительный и скучный код. Класс *itertools.product* возвращает объект генератора, и вместо того, чтобы сваливать на нас (а точнее, на наши Pi и Bramble) огромный список, он будет выдавать по одному паролю за итерацию. Параметр *repeat* — это длина выводимых строк. Таким образом, мы в цикле постоянно проверяем, не соответствует ли выходная свертка текущего пароля нашей целевой свертке.

```
for j in itertools.product(chars,repeat=length):
    guess = “.join(j)
    m = hashlib.md5(guess)
    hash = m.hexdigest()
    if hash == target_hash_md5:
        print “Победа!”
    break
return(guess)
```

Функции *md5cracker()* и *shacracker()* можно вызывать с этой длиной в качестве аргумента, и длину пароля легко изменить в соответствии с диапазоном длин для атак методом перебора. На одноплатном ноутбуке с процессором Athlon на 2,1 ГГц перебор всех паролей их трех символов занял около 0,7 секунды, из четырех — 40 секунд, а из пяти — около 40 минут. Так как в нашем диапазоне 62 символа, а скорость обработки по последнему показателю составляет около 380 тысяч сверток в секунду, то на пароль из шести символов понадобится около двух дней, а из восьми — около 18 лет. Для SHA-1 на том же компьютере скорость составила около 314 тысяч сверток в секунду, что примерно в 100 000 раз быстрее Pi. Компьютер поновее порадует вас больше, но каждый Pi по отдельности дает гораздо худший результат, выдавая жалкие 6 тысяч сверток в секунду для MD5 и немногим лучшие 8 тысяч сверток в секунду для SHA-1. И это без запущенного X-сервера и других демонов. В архитектуре ARM явно что-то не так — но дальнейшее исследование лежит за рамками нашей статьи, поскольку все, что можем делать мы, гораздо лучше делает *John the Ripper* (подробнее об этом позже). Кстати, в конце файла Python также содержится следующий блок кода:

```
if __name__ == ‘__main__’:
    import timeit
    print(timeit.timeit(stmt = “shacracker(4)”, setup=“from __
main__ import shacracker”, number=1))
```

Это означает, что если файл запускается из командной строки, он будет автоматически проанализирован модулем *timeit*. Кто-то попытается ускорить этот код с помощью Cython, преобразовав его в код C, но это может быть не слишком эффективно. Функции

hashlib и *itertools* написаны на C, и тут ловить особо нечего. Кроме того, даже если попробовать, у операций со строками в C есть собственные недостатки; если вы не владеете *malloc()* и *free()* в совершенстве, эти функции тоже вряд ли получится ускорить. Впрочем, попробуйте, будем рады.

Добавим мозгов

Перебор сверток — пример «до неприличия параллельных» вычислений: разделить задачу между узлами настолько просто, что не сделать этого прямо-таки нельзя. Поэтому вы можете смело умножать скорость обработки сверток на количество имеющихся у вас Pi, так как задача легко распределяется между ними путем простого изменения функции генератора. Если говорить конкретнее, то каждой Pi назначается фиксированный диапазон начальных символов пароля. Например, имея два Pi, одному из них можно оставить пароли, начинающиеся с первых 31 символа диапазона (0-ш), а второму — с символов остальной части диапазона (v-Z). При таком разделении никакого дальнейшего обмена данными между Pi не требуется — и им остается делать свою работу, не отвлекаясь на переговоры.

Весь параллелизм нам поддержит модуль *dispy*; для этого необходимо, чтобы все узлы запускали программу *dispynode*. После этого их можно будет легко призвать к действию. Мы разделили весь диапазон на 62 порции, которые узлы смогут запрашивать при необходимости. Это означает, что в кластер легко добавлять более быстрые (или медленные) компьютеры, иначе загрузку пришлось бы распределять вручную, чтобы избежать гонок.

Мы создаем собственный внешний цикл для первого символа, а все остальное внутри него делает цикл *itertools*.

```
def md5cracker(chunk_start,chunk_end,chars,length,thash):
    for j in range(chunk_start,chunk_end):
        for k in itertools.product(chars,repeat = length - 1):
            guess = chars[j] + “.join(k)
            ghash = hashlib.md5.new(guess).hexdigest()
            if ghash == target_hash_md5:
                return guess
    return False
```

Хотя вызывать эту функцию с параметрами подходящей порции можно с одного Pi на другом, распределение работы можно сделать менее нудным благодаря модулю *dispy*, который вы найдете на диске. Установите его командами

```
$ tar -xvzf dispy-3.15.tar.gz
$ sudo pip install -e dispy-3.15
```

Мы дали нашим Pi адреса 10.0.1.1 и 10.0.1.2. Это легко сделать, изменив файлы настройки сети или просто выполнив команду

```
$ ifconfig eth0 10.0.1.1/24
```

»

Жаргон распределенцев

Вычислительные архитектуры можно грубо разбить на четыре класса: «одна команда — одни данные (Single Instruction Single Data — SISD)», «одна команда — много данных (Single Instruction Multiple Data — SIMD)», «много команд — одни данные (Multiple Instruction Single Data — MISD)» и «много команд — много данных (Multiple Instruction Multiple Data — MIMD)». Это разделение возникло в 1960-х гг. и названо «таксономией Флинна» — по имени ее автора, Майкла Флинна [Michael J Flynn]. Как вы знаете, микропроцессорных систем тогда не существовало, но вычислительные машины тем не менее могли выполнять параллельные

операции, во многом по типу расширений Intel SSE (Streaming SIMD Extensions — Поточковые расширения SIMD), впервые появившихся в процессорах Pentium III. Сейчас некоторые приложения выходят за рамки классов Флинна, но те все равно довольно точно отражают ситуацию. Задача нашего урока — поиск исходного пароля — прекрасно входит в категорию SIMD: мы выполняем одну и ту же инструкцию на множестве наборов данных — дискретных наборах паролей заданной длины. Хотя ныне более всего распространена категория MIMD.

На одном краю спектра распределенных вычислений находится модель с распределенной

памятью, когда у каждого вычислительного модуля собственная память. Если процессоры должны обмениваться данными, нужно создать интерфейс обмена сообщениями. На другом краю — параллельные вычисления, когда есть общая память, доступная всем процессорам. Эта память может работать на основе шины, если все процессоры подключены к одной и той же плате, или программной, когда создается виртуальное хранилище памяти. В любом случае, конкуренция за доступ к шине и задержки при доступе означают, что работать с памятью при параллельных вычислениях нужно очень аккуратно.



Если у вас много Pi, придется подключить много кабелей для мониторов и клавиатур. Фактически адреса здесь не столь важны — если все узлы находятся в одной сети, *dispy* их обнаружит.

Простой кластер заданий для нашей функции *md5cracker()* с помощью *dispy* создается так:

```
cluster = dispy.JobCluster(md5cracker, callback=callback)
```

Параметр *callback* велит нашему кластеру вызывать функцию *callback()* каждый раз, когда модуль находит пароль или когда у него кончаются задачи. С помощью этой функции также можно объединять промежуточные или приблизительные результаты для более сложных сценариев, но мы пока не будем о них беспокоиться. Нашей функции нужно лишь проверить результат, и если он не *False*, можно завершить выполнение всех остальных задач:

```
def callback(job):
    if job.result:
        print "Great Victory!", job.result
    for j in jobs:
        if j.status in [dispy.DispyJob.Created, dispy.DispyJob.
Running]:
            cluster.cancel(j)
```

Для функции *md5cracker()* необходимы модули *itertools* и *hashlib*, и их нужно импортировать внутри функции. По соображениям синхронизации *dispy* не позволяет обмениваться внутренними переменными, поэтому функция *md5cracker()* принимает такие дополнительные параметры: *chars* (список символов), *length* (длину пароля) и *thash* (целевая свертка).

Наконец, воспользуемся следующим шаблонным кодом для запуска системы при запуске скрипта *multi.py* и отображения некоторой статистики, когда работа системы завершится:

```
if __name__ == '__main__':
    cluster = dispy.JobCluster(md5cracker, callback=callback)
```

```
jobs = []
start = 0
for j in range(nchunks):
    end = start + chunk_size
    job = cluster.submit(start, end, chars, length, target_hash_md5)
    job.id = j
    start = end
    jobs.append(job)
cluster.wait()
cluster.stats()
```

Не бойтесь Потрошителя

Существует масса альтернатив модуля *dispy* (<https://wiki.python.org/moin/ParallelProcessing>), но мы выбрали именно его, так как он хорошо подходит для реализации простого параллелизма, например, при распределенном вычислении свертки. Если вы хотите исследовать параллельную обработку на обычном многоядерном процессоре, обратитесь к «родной» библиотеке Python для параллельной обработки. Это позволит гораздо проще и надежнее работать с разделяемыми переменными, чем в случае, когда ваши процессы не локальны. При желании еще глубже нырнуть в кроличью нору параллелизма следует познакомиться с MPI (Message Passing Interface — Интерфейс передачи сообщений).

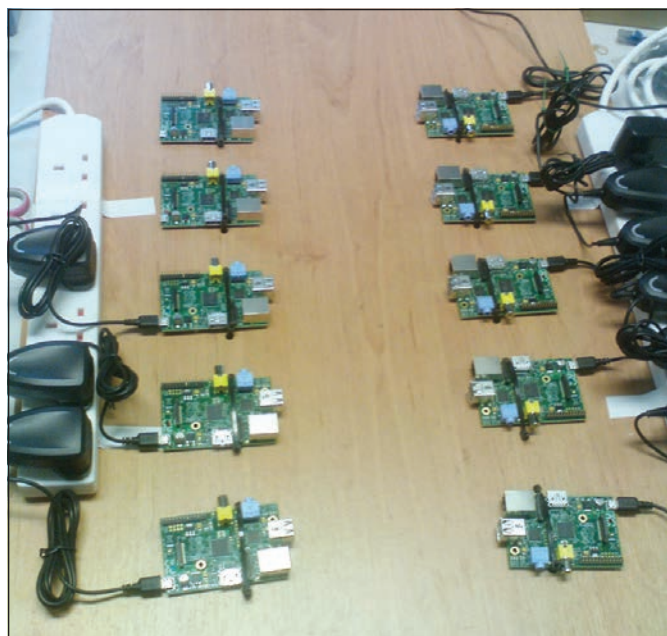
John the Ripper [«Джон-потрошитель»] — популярная открытая программа для взлома паролей. Еще есть улучшенная редакция программы от сообщества, которая поддерживает взлом с использованием графического сопроцессора, а также некоторые дополнительные типы свертки и шифров. Исходный код данной версии (называемой *Jumbo*) доступен на сайте проекта www.openwall.com/john. Ее можно скомпилировать из исходников на Pi. Raspbian по умолчанию включает *gcc*, но дополнительно

Другие проекты на Bramble

Взлом офисных документов (10 Raspberry Pi)

www.softwareontheside.info/2013/01/myraspberrypicluster.html

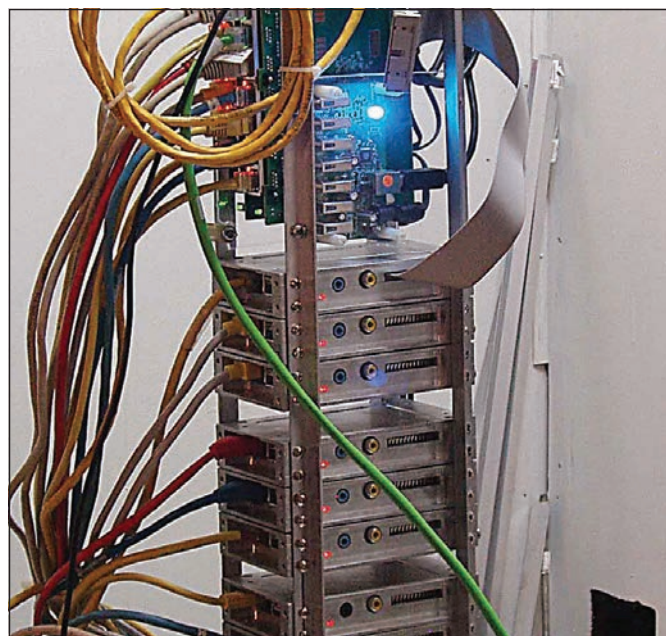
В проекте Шейна [Shane] используется измененная версия *John the Ripper* для взлома защищенных паролями документов *Microsoft Office* и *LibreOffice*, а также PDF. Со страху в *MS Office 2007* появилось шифрование AES со 128-битным ключом, упс!



Башня из Pi (10 Raspberry Pi)

<http://terminus.attitude.net/raspberrypi>

В проекте используется 9 подчиненных RPi и главный контроллер в виде платы Cubie для обработки данных с MPI. У устройства есть цветной дисплей 16×2 для отображения информации о состоянии. Теперь устройство используется в проекте Cubical Monolith из 32 узлов.



нужны заголовки *libssl* и *libcrypto*. Следующие команды установят их в Raspbian — для этого необходимо интернет-подключение:

```
$ apt-get update
$ apt-get install libssl-dev
```

Загрузите исходный код с сайта, распакуйте его и начните долгий процесс компиляции:

```
$ tar -xvzf john-$VER-jumbo-$REL.tar.gz
$ cd john/src
$ make generic
```

На стандартной Pi это займет около получаса. После окончания сборки можно проверить установку командами

```
$ cd ../run
$ ./john --test
```

Наш свежееизготовленный исходник теперь будет оценивать работу всех своих алгоритмов. Чистый MD5 получил около 330 тысяч сверток в секунду, что примерно в 50 раз быстрее нашей реализации на Python. Для SHA-1 (`--mode=rawsha1`) *John* выдал достойные 190 тысяч сверток в секунду. Теперь создайте текстовый файл **target.md5**, состоящий из одной строки с нашей целевой сверткой MD5, начинающейся с 392b2. Наш набор букв и цифр можно задать с помощью параметра `incremental` для множества символов Alnum.

```
$ ./john --incremental=alnum --length=6 --format=raw-md5 target.md5
```

Среднестатистический Pi провозится с этим около двух дней, и вы все равно не найдете исходный пароль. Эта задача остается вам, дорогие читатели. Но ее можно упростить, воспользовавшись невероятно полезным параметром *JtR* — `--node`, позволяющим достичь некоего примитивного параллелизма. Например, решив обогатить нашу систему пятью Pi, можно было бы дописать к предыдущему коду параметр `--node n/5`, заменив **n**

количеством Raspberry Pi, на которых выполняется код. С этим параметром *John* будет работать только с каждым соответствующим пятым значением, и мы сможем перебрать все буквенно-цифровые пароли из шести символов за 10 часов.


Для генерации сверток MD5 можно воспользоваться командой **md5sum**. Например:

```
$ echo -n «test»|md5sum
098f6bcd4621d373cade4e832627b4f6 -
```

Вставьте этот результат без пробела и дефиса в конце (*md5sum* чаще всего используется для определения свертки файлов, и за этими символами обычно следует имя файла — если они вам не нужны, добавьте `lcut -d -f1`) в файл **test.md5**. Теперь выполните следующую команду, чтобы убедиться, что *John* на самом деле способен корректно работать с этими свертками.

```
$ ./john --incremental=alnum --length=4 --format=raw-md5 test.md5
```

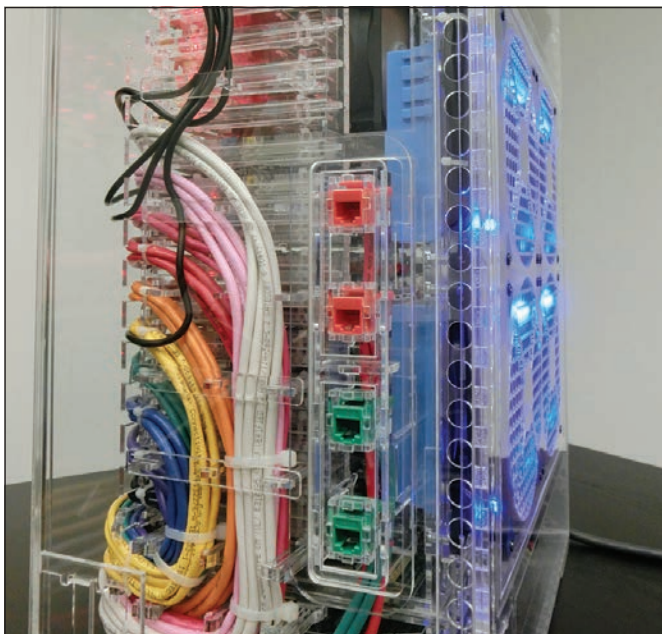
Слова и дело

Нельзя не упомянуть и о более интеллектуальном режиме *JtR* — проверке по списку слов. Несколько списков можно загрузить с сайта; также можно определить правила их сочетания. Существует и гибридный режим, с добавлением символов, что повышает ваши шансы успеть до кончины Вселенной. На момент написания статьи самый впечатляющий результат для MD5 близок к 2 миллиардам сверток в секунду (это достигается с помощью 16 видеокарт Radeon 7550). С такой скоростью пароль из восьми символов (с символами только из нашего диапазона в 62 символа) можно вскрыть за неделю. Более быстрые технологии всегда на подходе, поэтому старое утверждение, что пароль из 8 символов безопасен, более не актуально. Короче: заведите себе менеджер паролей и будьте осторожны, детки. 

Кластер Дэвида Гилла [David Guilli] на 40 Pi

<http://likemagicappears.com/projects/raspberrypicluster/>

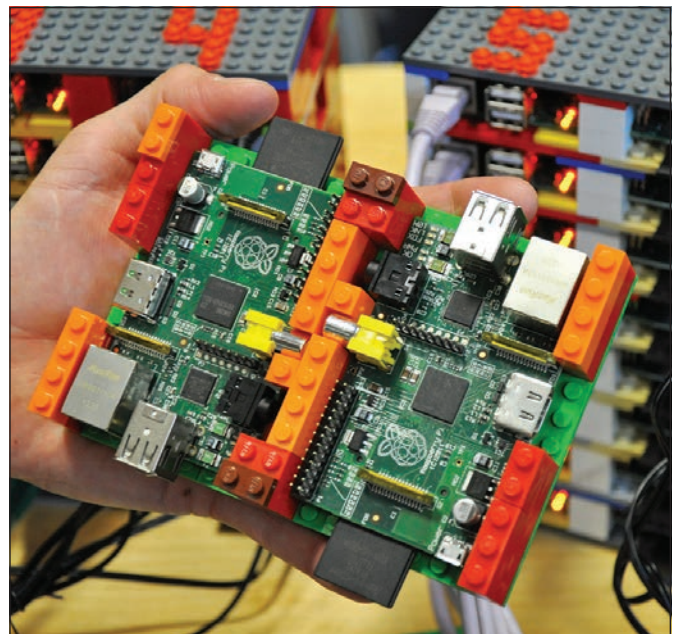
Задачей Дэвида было создать модель суперкомпьютера, которая бы полностью имитировала поведение настоящего. У этой зверушки 12 дисков по 1 ТБ и 6 внешних портов Ethernet. Стильный корпус был изготовлен с помощью лазерной резки с программным управлением.



Суперкомпьютер IridisPi Lego Университета Саутгемптона (64 Pi)

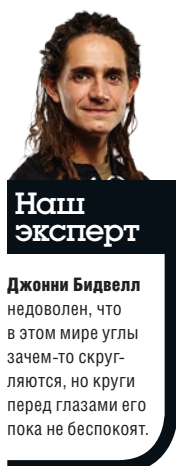
<http://www.southampton.ac.uk/~sjc/raspberrypi/>

Вся эта штука стоит всего £2500 и достигает производительности 1,14 ГФлоп/с при оценке с помощью теста на инверсию матриц HPL на всех 12 узлах. В ней используется более старая модель Pi с 256 МБ ОЗУ, так что память скудная.



Python: Пишем плагин для GIMP

Джонни Бидвелл использует Python, чтобы добавить новые возможности в любимый открытый графический редактор, даже не заикаясь о масках *GIMP*.



Наш эксперт

Джонни Бидвелл недоволен, что в этом мире углы зачем-то скругляются, но круги перед глазами его пока не беспокоят.



Пресечем нехорошие намеки [англ. *gimp* — в т.ч. «калека»]: *GIMP* умеет расширять свою функциональность с помощью плагинов. Желаящие взяться за дело не подетки могут писать такие плагины на C, пользуясь библиотеками *libgimp*, но это способно быстро обескуражить или взбесить. К счастью, для *libgimp* есть «мягкие» API, и с их помощью можно писать код на внутреннем языке *GIMP* — Script-Fu (на базе Scheme), на Tcl, Perl или Python. На нашем уроке мы рассмотрим последний язык в списке — самый доступный из них; даже если раньше вы вообще ничего ни на чем не писали, что-нибудь у вас да получится.

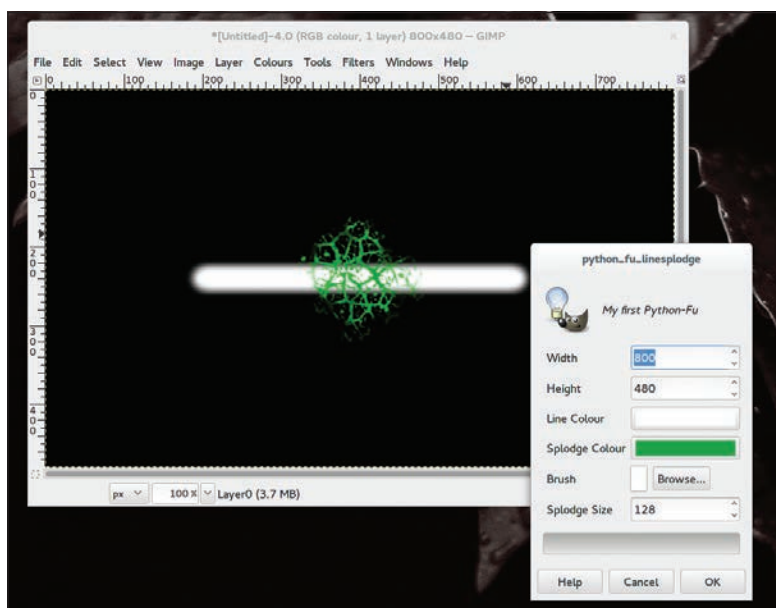
Начинаем

В Linux, к большинству пакетов *GIMP* прирана вся комплектация Python; у ваших друзей из Windows и Mac эти пакеты включены в саму программу начиная с версии 2.8. Для проверки запустите *GIMP* и проверьте наличие пункта Python-Fu в меню Filters [Фильтры]. Если его там нет, проверьте свою систему. Если есть, выберите его. Если все идет по плану, вы увидите призывного вида консоль со строкой приглашения (>>>), жаждущей вашего ввода. Все, что может делать *GIMP*, зарегистрировано в т.н. базе данных процедур (Procedure Database — PDB). Щелкнув по кнопке Browse [Обзор] вы увидите в окне консоли список этих процедур, их параметры и результаты. В Python к каждой из них можно обратиться через объект *pdb*.

В порядке изящного введения попробуем сделать пустое изображение с выбранным цветом фона. Для этого нужно создать изображение, добавить на него слой и затем отобразить на экране.

```
image = pdb.gimp_image_new(320,200,RGB)
layer = pdb.gimp_layer_new(image,320,200,RGB,'Layer0',100,RGB_IMAGE)
pdb.gimp_image_insert_layer(image,layer,None,0)
pdb.gimp_display_new(image)
```

Здесь мы воспользовались четырьмя процедурами: *gimp_image_new()*, параметры которой задают ширину, высоту и тип изображения (RGB, GRAY или INDEXED); *gimp_layer_new()*, которая работает с ранее заданным изображением и принимает ширину, высоту и тип данных, а также название слоя, его прозрачность и режим наложения; *gimp_image_insert_layer()*, которая фактически добавляет слой на изображение, и *gimp_display_new()*, которая выводит изображение на экран. Прежде чем что-либо делать с изображением, нужно добавить на него слои, поскольку изображение без слоев — объект неощутимый. Подробную информацию об этих процедурах вы найдете в браузере процедур [Procedure Browser] — наберите *gimp-layer-new* в строке поиска, и вам покажут все возможные режимы наложения. Учтите, что в Python дефисы в именах процедур заменяются символами нижнего подчеркивания, так как дефисы зарезервированы для операции вычитания. Однако в строке поиска дефисы допускаются.



➤ Линии и пятна можно менять как нравится, хотя, честно говоря, особой пользы от этого нет.



Рисуем линию

Все это здорово, но как бы что-нибудь нарисовать? Начнем с простой линии. Сперва выберите кисть и ее цвет, который будет хорошо смотреться на вашем фоне. Затем наберите в консоли

```
pdb.gimp_pencil(layer,4,[80,100,240,100])
```

Отлично — прекрасно отцентрованная линия, как будто нарисованная карандашом. Первый параметр функции `gimp_pencil()` — это слой, на котором мы рисуем линию. Точки задаются несколько странно: сначала указывается число координат, вдвое больше числа точек, так как у каждой точки есть координаты x и y ; затем указывается список вида $[x1, y1, \dots, xp, yp]$. Соответственно, в нашем примере мы рисуем линию от точки $(80,100)$ до точки $(240,100)$. Для выбора и изменения цвета, кистей и т.д. мы тоже пользуемся функциями PDB:

```
pdb.gimp_context_set_brush("Cell 01")
```

```
pdb.gimp_context_set_foreground("#00a000")
```

```
pdb.gimp_context_set_brush_size(128)
```

```
pdb.gimp_paintbrush_default(layer,2,[160,100])
```

Если у вас есть кисть "Cell 01", приведенный выше код нарисует зеленое пятно в центре области рисования — холста [canvas]. Если нет, возникнет сообщение об ошибке. Список всех доступных кистей покажет вызов `pdb.gimp_brushes_get_list()`. Кисти больше подходят для рисования таких причудливых пятен, чем карандаши, а заглянув в описание функции `gimp_paintbrush` в браузере процедур, вы увидите, что кистью можно рисовать

с градиентами и затенениями. Мы здесь обошлись настройками по умолчанию/текущими, для простоты. На диске вы найдете файл `linesplodge.py`, который зарегистрирует наш код как полноценный плагин *GIMP*, дополненный несколькими настройками.

На остальной части урока мы напишем чуть более сложный плагин для создания эффектов боке [bokeh] на ваших картинках. Слово «боке» происходит из японского и означает «размытие» или «туман», а в фотографии так называют эффекты расфокусировки, вызванные источниками света, находящимися вне глуби-

» Применение нашего плагина позволило получить приятный эффект боке в подчеркнутых частях фотографии.

«Наш плагин создаст слой с „кругами боке“ и слой с размытой копией изображения.»

ны резкости. Часто эти эффекты выглядят как равномерно окрашенные размытые следы округлой формы, напоминающие блики в объективе, в ярких участках изображения. Эффект, который вы получите в каждом конкретном случае, зависит от объектива и диафрагмы; в зависимости от конструкции вы можете получить многоугольное или пончикообразное боке. В нашем упражнении мы ограничимся круглыми боке.

Наш плагин заставит пользователя расположить источники света на своем изображении по некоторой линии, причем

»

» В следующем номере Исследуем новый язык Haskell, на основе PHP

Скорая помощь



Просмотрите залежи любительских плагинов в реестре плагинов GIMP на <http://registry.gimp.org>.

Изображение считается однослойным. Пользователь укажет диаметр окружности, радиус размытия, оттенок и насыщенность. В результате получатся два новых слоя: прозрачный верхний слой, содержащий «круги боке», и слой с размытой копией исходного изображения. Исходный слой остается нетронутым и находится под этими двумя. Меняя прозрачность двух новых слоев, можно достичь более интересного результата. Чтобы боке выглядело более реалистичным, часть изображения должна оставаться в фокусе и без кружков, поэтому удобно стереть некоторые части слоя, содержащего размытое изображение. Если пользователь не будет переименовывать слои, то при следующих запусках плагина новых слоев не добавится. Это означает, что можно применять функцию несколько раз с разными параметрами, и все круги боке все равно будут находиться на одном слое. Параметр размытия после первой итерации рекомендуется обнулить, иначе будет размываться уже размытый слой.

После инициализации необходимых переменных перейдем к созданию наших двух слоев. Для получения размытого слоя мы скопируем исходное изображение и добавим канал прозрачности. Слой боке создается почти так же, как в предыдущем примере.

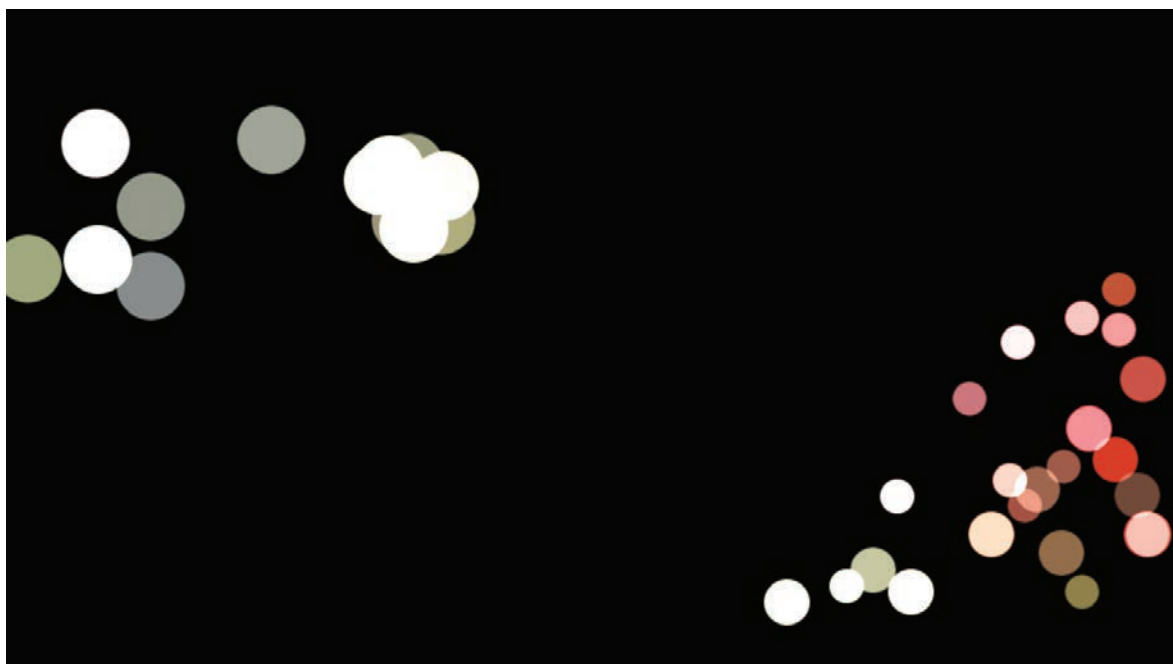
```
blur_layer = pdb.gimp_layer_copy(timg.layers[0],1)
pdb.gimp_image_insert_layer(timg, blur_layer, None, 0)
bokeh_layer = pdb.gimp_layer_new(timg, width, height,
RGBA_IMAGE, "bokeh", 100, NORMAL_MODE)
pdb.gimp_image_insert_layer(timg, bokeh_layer, None, 0)
```

Следующая задача нашего скрипта — извлечь список точек из выбранного пользователем пути. Это не слишком просто, так как общий путь может быть довольно сложным объектом: с кривыми, с изменением направления и т.п. Подробности описаны во врезке ниже, но сильно не утруждайтесь — вам достаточно понимать, что главный цикл **for** будет проходить в порядке рисования пути, извлекая координаты точки каждого компонента в виде двух переменных *x* и *y*.

После извлечения информации о точках наша следующая задача — получить локальный цвет изображения в этих точках. Функция PDB, которая это делает, называется **gimp_image_pick_color()**. У нее есть несколько параметров, отражающихся в диалоговом окне выбора цвета. В нашем случае мы получаем окружность с центром в точке (*x,y*) и радиусом 10 пикселей и вычисляем средний цвет для нее. Это более предпочтительно, чем возвращать цвет одного пикселя в точке (*x,y*), так как цвет этого пикселя может не совпадать с окружающими.

Поливаем из ведра

Рисование круга выбранного цвета на слое боке начнем несколько неинтуитивно — с рисования черного круга. Вместо того, чтобы воспользоваться кистью, для которой у всех пользователей должны быть одинаковые наборы цветов, нарисуем круг, просто залив выделение в форме окружности. Для выделения воспользуемся командой



› Вот наши круги. При желании можно смешать цвета или добавить градиенты, но и одноцветные круги выглядят замечательно.

Пути, векторы, линии, точки, изображения и области

Пути хранятся в объекте, называемом «вектор [vector]». Точнее, объект содержит набор линий, каждая из которых описывает часть пути. Мы предполагаем, что путь простой и не содержит кривых, поэтому у нас будет всего одна линия, из которой мы выудим наши желанные точки. В коде эта линия называется **gpoints**. Эта переменная фактически представляет собой массив, третьим элементом которого является список точек. Так как массивы в Python нумеруются с 0, то список точек соответствует **gpoints[2]**. Этот список имеет форму `[x0,y0,x0,y0,x1,y1,x1,y1,...]`.

Каждая точка здесь фигурирует дважды, так как при других параметрах список должен содержать информацию о кривых. Чтобы избежать повторов, мы используем 4 в качестве шага функции **range()**, так мы получим значения *x* 0, 4, 8 и значения *y* 1, 5, 9. Длина списка точек передается нам во втором элементе **gpoints**.

```
for j in range(0,gpoints[1],4):
```

В коде вы увидите несколько обращений к переменным **timg** и **tdraw**. Это изображение и слой (точнее, изображение и области рисования

[drawable]), активные на момент вызова функции. Как вы понимаете, иметь их под рукой довольно удобно, так как очень многим функциям нужны по крайней мере изображение и слой для работы. По сути, даже настолько удобно, что при регистрации своего скрипта в GIMP упоминать о них не придется — предполагается, что вы непременно передаете их своей функции. Класс под названием **drawables** образуют слои и каналы — обобщение здесь вполне оправдано, так как много чего одинаково применимо и к тому, и к другому.

```
pdb.gimp_image_select_ellipse(timg, CHANNEL_OP_REPLACE, x
- radius, y - radius, diameter, diameter)
```

Существует несколько констант для выбора различных сугубо *GIMP*овских режимов и прочих таинств. Их легко определить по «кричащему» верхнему регистру. Второй аргумент здесь означает число 2, а также тот факт, что текущее выделение следует заменить на указанное эллиптическое.

Размеры выделения задаются через указание левого верхнего угла прямоугольника, в который вписан эллипс, и ширины этого прямоугольника. Мы срежем края этого выделения на два пикселя и затем установим фоновый цвет в черный. Затем зальем это новое выделение в режиме Сзади [Behind mode], чтобы не повлиять на другие круги на слое:

```
pdb.gimp_selection_feather(timg, 2)
pdb.gimp_context_set_foreground('#000000')
pdb.gimp_edit_bucket_fill_full(bokeh_layer,0,BEHIND_
MODE,100,0,False,True,0,0,0)
```

И вот почему мы сделали круг черным: мы будем рисовать круги в режиме со смешением цветов. Это означает, что области наложения кругов станут светлее, по слабой аналогии с фотографией. Проблема в том, что смешение цветов никак не влияет на прозрачность, поэтому сначала мы делаем круг черным, а потом весь черный цвет сменится нашим новым цветом.

```
pdb.gimp_context_set_foreground(color)
pdb.gimp_edit_bucket_fill_full(bokeh_layer,
0,ADDITION_MODE,100,0,False,True,0,0,0)
```

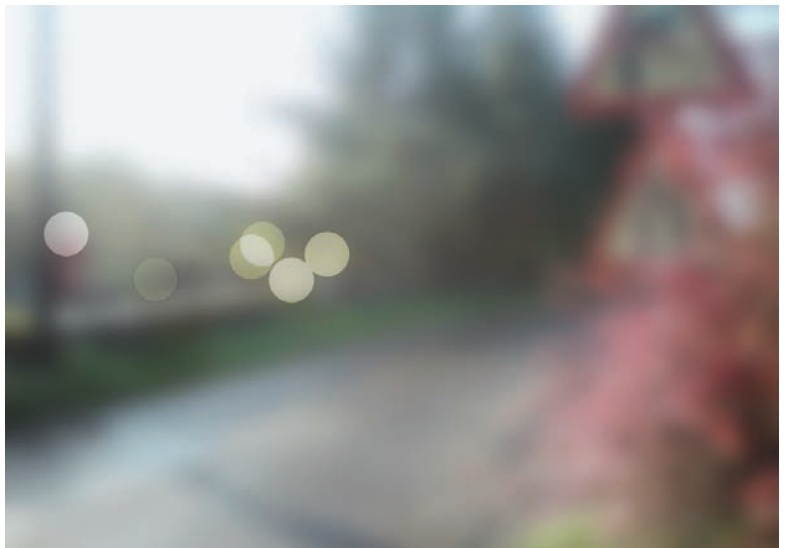
Нарисовав таким образом все круги, при необходимости выполним на скопированном слое гауссово размытие. Мы сказали, что часть изображения должна остаться в фокусе — поработайте с этим слоем позже, чтобы он стал более прозрачным в нужных областях. Перед заливкой нужно снять выделение, так как иначе мы сделаем размытым только что нарисованный круг.

```
if blur > 0:
pdb.plug_in_gauss_iir2(timg, blur_layer, blur, blur)
```

Мягче, мягче

Наконец, применим наши установки оттенка и интенсивности цвета и переведем слой боке в режим мягкого света [Soft-Light mode], чтобы нижние слои под кругами подсвечивались. И на всякий случай, если какой-то круг избежал заливки, польем его с помощью плагина Color-To-Alpha.

```
pdb.gimp_hue_saturation(bokeh_layer, 0, 0, lightness,
saturation)
pdb.gimp_layer_set_mode(bokeh_layer, SOFTLIGHT_MODE)
pdb.plug_in_colortoalpha(timg, bokeh_layer, '#000000')
```



Таковы общие функции скрипта. По коду на диске вы увидите, что мы сделаем еще кое-что, в частности, сгруппируем всю серию операций в одну неотменяемую и сбросим все новые настройки инструментов, проставленные скриптом. Всегда стоит прибавить после себя, оставив все как было до вас! В функции `register()` мы задаем ее путь в меню `<Image> > Filters > My Filters > PyBokeh...`, чтобы она правильно зарегистрировалась в подменю `My Filters` [Мои фильтры] меню `Filters` [Фильтры]. В это подменю вы сможете добавить новые скрипты, чтобы не забывать ими и без того переполненное меню `Filters`. На примерах изображений показаны несколько результатов применения плагина `PyBokeh`.

➤ После применения фильтра все немного размывается. Изменение прозрачности слоя позволяет вернуть некоторые детали.

«Сгруппируем операции в одну неотменяемую и сбросим новые настройки инструментов.»

Критиканы могут празднично глумиться насчет полезности скрипта, и действительно: все, что он делает, можно сделать вручную, да еще и лучше. Это, в той или иной степени, справедливо для любого скрипта *GIMP*. Но делать такое вручную очень утомительно и может привести к ошибкам — нужно фиксировать координаты и цвет центра каждого круга и очень искусно накладывать круги, если хотите сохранить смешение цветов. **LXF**

Регистрация плагина

Чтобы ваш плагин появился в меню *GIMP*, его нужно определить как функцию Python и затем воспользоваться функцией `register()`. Самый элегантный вариант — сохранить весь код в скрипте Python определенного вида. Общая форма такого скрипта следующая:

```
#!/usr/bin/env python
from gimpfu import *
def myplugin(params):
# далее идет код
register(
proc_name, # например, "python_fu_linesplodge"
blurb, # "Рисует линию и кляксу"
help, author, copyright, date,
menupath, imagetypes,
params, results,
```

```
function) # "myplugin"
main()
```

Параметр `proc_name` определяет название нашего скрипта в PDB; в начале этого названия автоматически добавляется `'python_fu'`, чтобы у всех плагинов Python была собственная ветвь в таксономии функций. Параметр `menupath` определяет вид регистрируемого плагина и где плагин появится в меню *GIMP*: в нашем случае достаточно будет `<Image> > Filters > Artistic > LineSplodge...`. `imagetypes` определяет тип изображений, с которыми работает плагин, например, `"RGB", "GRAY"` или просто `"",` если он не работает ни с какими изображениями, как в нашем примере. Список `params` задает элементы управления вводом исходных данных плагина: для них можно использовать удобные

интерфейсы, задаваемые специальными типами Python-Fu, например, `PF_COLOR` и `PF_SPINNER`. Список `results` описывает результат работы плагина, если он есть. В нашем случае будет достаточно (`PF_IMAGE, image, "LSImage"`). Наконец, `function` — это просто название нашей функции в Python, под которым она появляется в коде.

Чтобы *GIMP* нашел и зарегистрировал ваш плагин при последующей загрузке, сохраните созданный файл под именем (например) `myplugin.py` в каталоге плагинов: `~/gimp-2.8/plugin-ins` для Linux (сделайте файл исполняемым командой `chmod +x myplugin.py`) или `%USERPROFILE%\gimp-2.8\plugin-ins\` для Windows, при необходимости заменив номер версии *GIMP* на тот, что установлен в вашей системе.

Fortran: Давайте приступим

Fortran — язык далеко не юный, но все еще имеет хождение, и Джульетта Кемп считает, что он проще, чем можно предположить по его репутации.



Наш эксперт

Писатель Джульетта Кемп полагает, что есть нечто странно-приятное в работе с языком, которым пользовался еще ее отец. Хотя ее радует, что прогресс вывел из оборота перфокарты.

Эта программа f95 просит вас ввести свое имя и приветствует вас; затем идет версия Hello World для F77 (подробнее о фиксированном формате см. во врезке).

По любым стандартам Fortran — старый язык (загляните во врезку «Версии Fortran» внизу этой страницы, и вы узнаете о его истории). Но, несмотря на возраст, он по-прежнему весьма активно применяется в точных науках, в высокопроизводительных вычислениях и в суперкомпьютерах (это лишь несколько примеров); и хотя кое-кто уже не одно десятилетие предсказывает его неминуемую гибель, он пока что крепко. Область, где Fortran особенно хорош — численные расчеты, и его нелегко потеснить с его сильных позиций в инженерии,

физике и математике. Кроме того, он используется рядом инвестиционных банков и страховых компаний в сложных математических моделях.

Коду Fortran уже 35 лет, но он по-прежнему в седле и в некоторых областях активно разрабатывается. Конечно, если вы занимаетесь web-дизайном или GUI приложений, то, скорее всего, захотите чего-нибудь другого. Но, как и со всеми языками данной рубрики, всегда интересно копнуть поглубже и посмотреть, что у вас получится.

Начинаем

Возвращаясь к врезке о версиях Fortran — для нашего урока мы примем Fortran 90, и это означает необходимость найти компилятор Fortran 90, который вам понравится. Проще всего, вероятно, будет с компилятором от GNU, *GFortran*. Среди других

вариантов — *G95* (бесплатный) и *NAG* (платный). В некоторых компиляторах есть дополнения к стандарту Fortran, которые бывают полезны, но привязывают вас именно к этому компилятору; разумнее будет придерживаться официального стандарта.

На нашем уроке я воспользуюсь *GFortran* — этот компилятор имеется на сайте GNU или в виде пакета для большинства дистрибутивов. В Debian можете запустить его командой **gfortran** или **f95** (обе запустят одно и то же); я буду употреблять в тексте **f95**.

Откройте файл **hello.f95** и введите следующий код:

```
! Hello World
program hello
print *, "Hello World"
end program hello
```

Скомпилируйте его с помощью **f95 -o hello hello.f95**, затем запустите с помощью **./hello** (попробуйте сделать это без **-o hello**, и вы увидите, что готовый файл на выходе по умолчанию именуется **a.out**. Он будет работать нормально, но довольно неудобно не знать,

```

juliet@inspiral: ~/coding/fortran
File Edit View Search Preferences Tabs Help
1. juliet@inspiral: ~/coding/fortran x
juliet@inspiral:~/coding/fortran$ f95 -o hello hello.f95
juliet@inspiral:~/coding/fortran$ ./hello
Please enter your name:
Juliet
Hello Juliet
juliet@inspiral:~/coding/fortran$ f95 -o hellofor hello.for
juliet@inspiral:~/coding/fortran$ ./hellofor
Hello World
juliet@inspiral:~/coding/fortran$ cat hello.for
*
Hello World
program hello
print *, 'Hello World'
end program hello
juliet@inspiral:~/coding/fortran$

```

Версии Fortran

Fortran (или, скорее, его первый рабочий компилятор) появился в 1957 году в качестве альтернативы языку ассемблера для программирования на больших ЭВМ. Тогда он содержал аж 32 утверждения и хранился, естественно, на перфокартах, по одной на строку. Он быстро обрел популярность — да и кто бы не предпочел его ассемблеру? Поэтому конкурирующие производители моментально принялись разрабатывать компиляторы для разных типов мейнфреймов. Таким образом, это бесспорно

первый кросс-архитектурный язык программирования. В следующие пару лет появились Fortran II, III и IV, но большим шагом вперед стал Fortran 66: эта версия языка превратилась в «промышленный стандарт».

Fortran 77, десять лет спустя, исправил недочеты F66; это, вероятно, самый исторически значимый диалект. Вы до сих пор можете найти компиляторы F77 для современных компьютеров, и F77 до сих пор существует в природе, делая свое дело.

Важной вехой стал долгожданный Fortran 90. Однако он сохранил все основные функции F77, и любая программа, совместимая с F77, должна быть также совместима с F90. Fortran 95 был менее существенным пересмотром.

С тех пор опубликован другой важный пересмотр — Fortran 2003 (и мелкое обновление в 2008). Но если вы хотите освоить Fortran сегодня, лучше начать с 90/95, а затем изучать функции, добавленные в 2003. На нашем уроке применяется Fortran 95.

Свободный формат против фиксированного

До Fortran 90, код Fortran должен был писаться в фиксированном формате. Это уходило корнями в дни перфокарт, и правила были таковы:

- » Максимальная длина строки — 72 символа.
 - В знак продолжения строки, в колонке 6 следующей строки ставился произвольный символ.
 - » Первые 6 колонок должны были быть пустыми, если вы не использовали одну из них для знака комментария или символа продолжения строки.
 - » Строки комментариев имели в первой колонке * или c.
 - » Пробелы игнорировались.
- Наша простейшая программа “hello” выглядела бы таким образом:

```
c Hello World
program hello
  print *, 'Hello World'
end program hello
```

В F90 допускается свободный формат. Весь код нашего урока написан в свободном формате. Но все же некоторые правила имеются:

- » Строки не должны превышать 132 символа; используйте & для продолжения строки. Если разрыв — в середине имени, используйте еще один & в начале следующей строки:

```
character :: name*100
name = "это очень длинное имя,
поэтому понадобится амперсанд &
```

```
&и здесь тоже”
! comment
» Комментарии начинаются с !.
» Пробелы учитываются; нельзя их встраивать в имя переменной или в число. В ряде терминов, типа END IF, пробел нельзя опускать.
» Можно писать по несколько утверждений в одной строке, разделяя их ;. Отступы игнорируются.
```

Компилятор GCC по умолчанию считает формат свободным, если имя файла исходника оканчивается на .f95 или .f90, и фиксированным — для исходных файлов с .for. Вы можете указать в командной строке **-fixed-form** или **-free-form**, для отмены этих настроек.

какому исходному файлу он соответствует, и его очень легко случайно затереть). У вас должен получиться традиционный вывод **Hello World**.

Наши несколько строк показывают основные функции языка. ! используется для комментариев. Программы начинаются с **program <имя>** и заканчиваются на **end program <имя>**; имя программы не обязано соответствовать имени файла. Отступ не требуется, но его неплохо добавлять для лучшей читаемости кода.

print выводит на экран (звездочка означает «использовать соответствующий формат вывода для того, что выводится»). В F77 нужно было использовать одиночные кавычки, но в F90 двойные кавычки тоже вполне допустимы.

Основы языка

Давайте попробуем немного более сложную версию **hello.f95**:

```
! Hello Name
program hello
  implicit none
  character :: name*20
  print *, "Пожалуйста, введите свое имя: "
  read *, name
  print *, "Hello ", name
end program hello
```

Строка **implicit none** очень важна. Fortran старой школы (т.е. версия F77 или раньше) имел неявную типизацию переменных.

Переменные, имена которых начинались с i, j, k, l, m или n, считались целочисленными (этот язык писали ученые, которые автоматически воспринимали i, j и т.д. как целые числа), а все прочие имена были для вещественных чисел. Строковые переменные или символы должны были описываться явно. F95 такое сохранил (для обратной совместимости), но это отличный способ надеяться избежать ошибок. Утверждение **implicit none** не разрешает ничего подразумевать и заставляет описывать все переменные явно. Здесь мы заявляем, что **name** — это массив символов длиной 20; т.е. это строковая переменная из 20 символов (если вам хочется получить возможность работать с более длинными именами, увеличьте это число).

read, что вполне очевидно, считывает информацию с экрана в переменную. Скомпилируйте и запустите все, и у вас спросят ваше имя, а затем поприветствуют.

Обратите внимание, что базовые типы данных в Fortran — **INTEGER**, **REAL** (точность — 6 десятичных знаков), **DOUBLE PRECISION** (точность — 13 десятичных знаков), **LOGICAL** (булево; **.TRUE.** или **.FALSE.**), **COMPLEX** (комплексные числа) и **CHARACTER** (как показано выше, массив из **CHARACTER** — это строковая переменная).

При обсуждении кода часто используются сплошь заглавные буквы, но они не обязательны при его написании. Fortran 90 также ввел производные типы, позволяющие создавать более сложные структуры данных.

Скорая помощь

Если вам когда-нибудь придется заниматься отладкой чужого кода на Fortran, прогон его через разные компиляторы (со включенными функциями отладки) может пролить немало света на возможные проблемы.

Вычисления и циклы

Поскольку Fortran сильно математизирован, давайте создадим математическую программу. Вероятно, вы встречались с рядом Фибоначчи: это последовательность чисел, где каждый член является суммой двух предыдущих. Ряд Фибоначчи проявляется в самых разных областях, как в математике, так и в природе. Поскольку алгоритм его вычисления является итеративным (вы рассчитываете значение, потом используете его для получения следующего), для его генерирования в нашем коде мы воспользуемся циклом.

Вот небольшая программка, которая создает 20 первых чисел Фибоначчи:

```
! Вычисление ряда Фибоначчи до 20-го члена
program fibonacci
  implicit none
  integer :: first, second, a, i
  first = 1
  second = 1
```

```
print *, first
print *, second
do i = 1, 18, 1
  a = first + second
  print *, a
  first = second
  second = a
end do
end program fibonacci
```

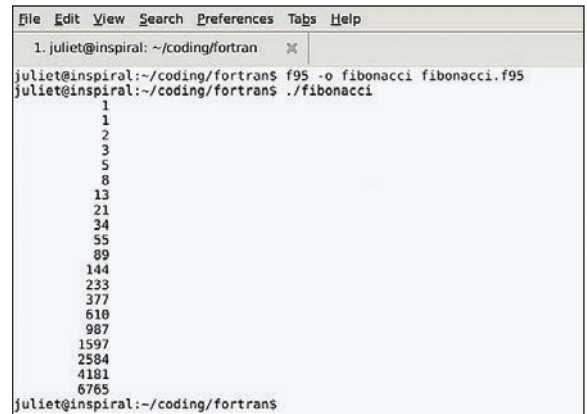
Мы начинаем с того, что задаем все нужные нам переменные: переменные для хранения первых двух чисел в последовательности (обе равны 1); **a** будет нашим временным числом при создании последовательности; **a i** — переменной цикла (итератором). Затем мы выводим два первых числа. Самое интересное — это цикл **do**. В F77 и более ранних версиях для создания цикла приходилось использовать метку и переход к ней по **GOTO**, но в F90 появился более современный синтаксис:

»

```
do iterator = start, end, increase
! Здесь идет тело цикла
end do
```

Итератор начинается со **start**, увеличивается на **increase** и завершается, когда итератор доходит до значения **end**. И так, у нас *i* увеличивается на 1 каждый цикл от 1 до 18. Поскольку первые 2 значения ряда у нас уже есть, это дает нам результат из 20 значений. Тело цикла **do** довольно простое; мы генерируем и вводим следующее значение в последовательности, затем переназначаем наши переменные **first** и **second**, готовя их к следующему циклу. Примечание: поскольку увеличение на 1 — случай очень частый, вы можете использовать **do i = 1, 18**; *i* будет увеличиваться на 1 по умолчанию.

Можете слегка изменить эту программу, чтобы она распечатывала в точности все, что делает в цикле, если вы не совсем уверены в работе алгоритма.



» Создание последовательности Фибоначчи.

Вычисление корня функции

Чтобы продемонстрировать более сложные функции языка, давайте возьмем более сложную математическую задачу: поиск корня функции.

Поиск корня функции означает, что для функции $f(x)$ нужно найти значение x , для которого $f(x) = 0$. Вот несколько примеров:

- » $f(x) = x - 2$: корень этой функции — 2, поскольку $2 - 2 = 0$.
- » $f(x) = 3x - 3$: корень этой функции — 1, поскольку $3 * 1 - 3 = 0$.
- » $f(x) = 2x^2 - 8$: корень этой функции — 2, поскольку $2 * 2^2 - 8 = 0$.

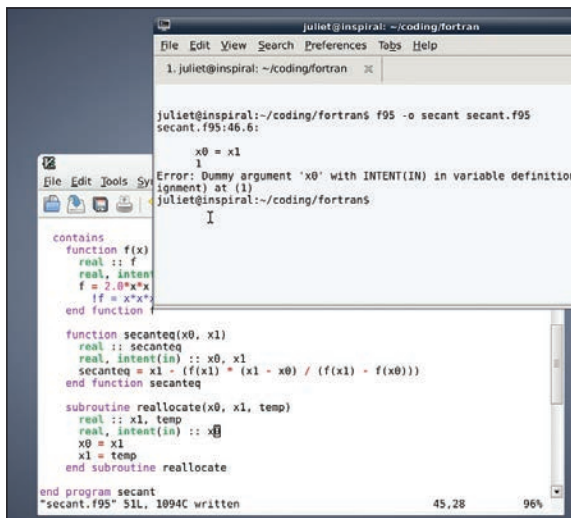
Это проявляется во многих математических задачах и ситуациях. Для всех приведенных функций решение можно найти

по формуле (как вы, вероятно, и делали в школе). Однако для поиска корня более сложных функций потребуется вычислительный подход.

Простой и очень старый способ (даже старше, чем Fortran; лет ему этак 3000) именуется методом секущей. Я не буду вдаваться в подробности работы данного метода (загляните в Википедию или еще куда-нибудь, если вам нужно исчерпывающее объяснение), однако реализация нижеприведенного алгоритма работает!

Сохраните в файле **secant.f95** следующий код:

```
! Вычисление корня 2x^2-1 методом секущей
program secant
implicit none
real, parameter :: error = 1.0e-6
real :: x0, x1, temp
integer :: i = 1, limit = 100
x0 = 0
x1 = 1.0
do
if (i > limit) then
print *, "Останавливаемся; нет сходимости"
exit
endif
if (f(x1) == 0) then
print *, "Корень равен ", x1
exit
else if (abs(x1 - x0) < error) then
print *, "Корень примерно равен ", x1
exit
endif
temp = secanteq(x0, x1)
```



» Здесь я использовала ключевое слово **intent(in)** для переменной, которую хочет изменить подпрограмма (показано в редакторе). Отладчик выдал ошибку.

Отладка

К сожалению, сообщения компилятора Fortran не всегда в полной мере полезны. Если вы получаете странные сообщения, обязательно вставьте **implicit none**. Можно также везде добавить утверждения **print** — и вам будет видно, что происходит.

После этого можете попробовать некоторые опции отладчика **GFortran**:

» **-O0** велит отладчику не делать оптимизации. Оптимизация способна вытворять странные штуки с локальными переменными, затрудняя отладку.

Недостаток в том, что это сильно тормозит работу вашего кода. Еще одна опция в более новых версиях **GFortran** (после 4.8) — **-Og**; она применяет только оптимизацию, не влияющую на отладку.

» **-Wall** активирует "all [все]" предупреждения;

-Wextra активирует только некоторые, не включенные в "all".

» **-pedantic** предупреждает о языковых функциях, которые не поддерживаются в **GFortran** и не входят в официальный стандарт.

» **-g** позволяет запустить ваш код через **GDB**, интерактивный отладчик GNU. Используйте ее с **gdb <имя программы>**; затем можете ввести **break main**, чтобы задать контрольную точку перед первым исполняемым утверждением, **run** для запуска контрольной точки и **step** для прохождения утверждения по шагам.

У **GDB** имеется обширная онлайн-документация, но для отладки Fortran существуют программы и получше.

```

call reallocate(x0, x1, temp)
i = i+1
!раскомментируйте строку print - увидите цикл в действии
!print *, "x0 is ", x0, "x1 is ", x1
end do
contains
function f(x)
  real :: f
  real, intent(in) :: x
  f = 2.0*x*x - 1.0
end function f
function secanteq(x0, x1)
  real :: secanteq
  real, intent(in) :: x0, x1
  secanteq = x1 - (f(x1) * (x1 - x0) / (f(x1) - f(x0)))
end function secanteq
subroutine reallocate(x0, x1, temp)
  real :: x0, x1, temp
  x0 = x1
  x1 = temp
end subroutine reallocate
end program secant

```

Прежде всего, рассмотрим общую структуру программы. Внутри программы, обрешенной строками **program ... end program**, есть подготовительный раздел, цикл **do**, и затем раздел **contains**.

Раздел **contains** позволяет создавать внутренние подпрограммы (функции или подпрограммы) внутри вашей программы. Можно также создавать внешние подпрограммы — они появляются после строки **end program**. Между функцией и подпрограммой два основных различия:

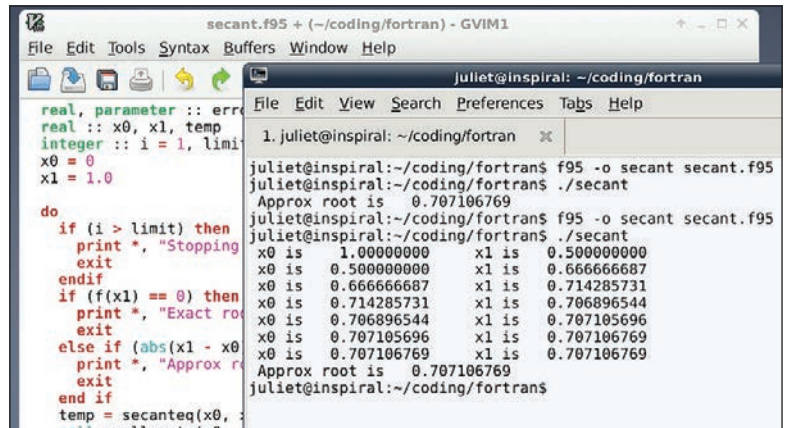
- » Функция возвращает значение; подпрограмма — нет.
- » В функции обязательно должен быть хотя бы один аргумент; в подпрограмме их может не быть.

Посмотрим, что у нас делают функции и подпрограммы. На данный момент достаточно обратить внимание, что они определены ключевыми словами **function ... end function** и **subroutine ... end subroutine**.

Взгляните, что происходит в программе. Первые несколько строк задают всякие переменные. Единственная новость здесь — это ключевое слово **parameter**. В Fortran оно используется для определения постоянной. Объявляя параметр, вы принимаете, что значение параметра не будет изменяться в течение программы (и вы получите ошибку отладчика, если попытаетесь его изменить). Мы также задаем **x0** и **x1** — это будут наши итеративные значения (так же, как вы видели в программе Фибоначчи). Идея здесь в том, что по мере повторений эти значения все ближе подбираются к настоящему корню уравнения.

И снова, большая часть логики программы находится в цикле **do**. На сей раз переменной цикла у нас нет, а цикл бесконечный. Если вы такой употребили, надо позаботиться о наличии условий останова. В данном примере мы сравниваем **i** (используемый как неформальный итератор) с **limit**, и выходим из цикла. Ключевое слово **exit** выводит вас из цикла, в котором оно находится. Каждый прогон цикла смотрит на наши два текущих значения **x** и или выходит из цикла, или пересчитывает их.

Если 100-раундовый лимит пока не достигнут, мы тестируем самое недавнее значение **x**, **x1**, чтобы проверить, вернет ли оно 0 при включении в функцию. Если да, то мы нашли корень и можем выйти. Мы также делаем проверку на «почти равно»; если два самых недавних значения **x**, **x0** и **x1** имеют очень незначительное различие, значит, мы получили приблизительное значение корня и тоже можем остановиться. Переменная **error** задает точность; здесь она проверяет, чтобы разница между значениями составляла менее 0,000001. Если ни одно из этих условий



не выполняется, мы продолжаем искать корень. Таким образом, мы вычисляем уравнение секущей для **x0** и **x1** и сохраняем его в **temp**. Затем подпрограмма **reallocate** задает старый **x1** в качестве нового **x0**, а **temp** в качестве **x1**; и мы снова возвращаемся к началу цикла.

Подпрограммы

У нас в разделе **contains** есть две функции. Одна, **f(x)**, это функция, корень которой нам надо найти. Другая — **secanteq(x0, x1)**, которая применяет метод секущей к нашим двум рабочим значениям. Внутри функции, имени функции присваивается тип, и она начинает считаться переменной. Это значение, которое будет возвращено из функции, и ее тип вы должны объявить явно, как здесь. Вы также должны указать тип аргументов. Попытавшись передать переменную, не соответствующую типу аргумента, вы получите ошибку компиляции.

Еще одно новшество — ключевое слово **intent(in)**. Оно помогает избежать случайного редактирования или передачи значений из функции. Теперь эти значения определяются как **input-only**, и если функция попытается их изменить, при компиляции появится сообщение об ошибке.

В подпрограмме аргументы тоже должны иметь определенный тип. Однако, в отличие от функции, значения подпрограмма не возвращает. Вместо этого она изменяет любые значения внутри себя (если вы передаете ей значения, которые не хотите изменять, можете снова использовать **intent(in)**). Мы жонглируем тремя значениями: сбрасываем старое **x0**, старое **x1** становится **x0**, а новая временная переменная становится **x1** (если хотите, можете делать это методом встраивания; здесь я частично его раскрыла, чтобы продемонстрировать подпрограмму). Собрав все это вместе, вы теперь можете скомпилировать программу и запустить ее — и посмотреть, получите ли вы ответ.

Этого должно быть достаточно, чтобы познакомить вас с тем, как работает Fortran. Если вы привыкли к более современным языкам, он может показаться вам несколько неуклюжим. Однако он отлично подходит для определенных числовых вычислений; и есть огромные куски кода Fortran в свободном доступе для решения многих математических и научных задач. Если вам понадобится код Fortran, сначала проверьте онлайн-библиотеки, и только потом беритесь за собственный. Fortran 95 позволяет использовать модули и другие структуры, которые упрощают большие программы, но это не входит в задачи нашего урока.

И, наконец, привлекаем ваше внимание к *Fortran Colouring Book* [Книга-раскраска по Fortran'у] Роджера Кауфмана [Roger Kaufman] — это, вероятно, самое интересное руководство по компьютерному языку из всех имеющихся (книга рукописная, со множеством иллюстраций в стиле Доктора Сьюза). К сожалению, поскольку написана она в 1978 году (как учебник MIT), она охватывает только F77, но не F95. [LXF](#)

» Поиск корня уравнения. Можете раскомментировать строку печати в цикле, если хотите понаблюдать за сходимостью значения, как здесь.

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 Установку `yEd` в Wheezy
 - 2 Grub и MS-DOS
 - 3 Воспроизведение видео в Chrome
 - 4 Воспроизведение DVD
- 5 Незагружаемые DVD
 - 6 Беспроводное подключение в Xubuntu

1 Java и путаные пути

Вя пытаюсь установить 32-битную версию `yEd` в свою Debian Wheezy, и каждый раз все заканчивается следующей ошибкой:

```
./yEd-3.12.2_32-bit_setup.sh
```

```
Unpacking JRE ...
```

```
Preparing JRE ...
```

```
bin/unpack200: error while loading shared libraries:  
libgcc_s.so.1: cannot open shared object file:
```

```
No such file or directory [Невозможно открыть  
разделяемый файл]
```

```
Error unpacking jar files. The architecture or bitness  
(32/64) of the bundled JVM might not match your  
machine [Ошибка распаковки файлов. Архитектура  
или разрядность JVM может не соответствовать  
вашей машине].
```

После продолжительного исследования я нашел несколько сообщений о проблемах, связанных с попытками установки 32-битной виртуальной машины Java (JVM) на 64-битный компьютер. Везде рекомендуется установить пакет `ia32-libs`; но, попробовав установить этот пакет в своей системе, я столкнулся с проблемой. Другой вариант — попробовать выполнить установку вручную, как описано на <http://yed.yworks.com/support/qa/17>. Там рекомендуется «набрать в терминале

```
java -jar /path/to/yed.jar
```

(заменяя `/path/to/` фактическим путем); эта команда запустит `yEd`. Запуск простого скрипта

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

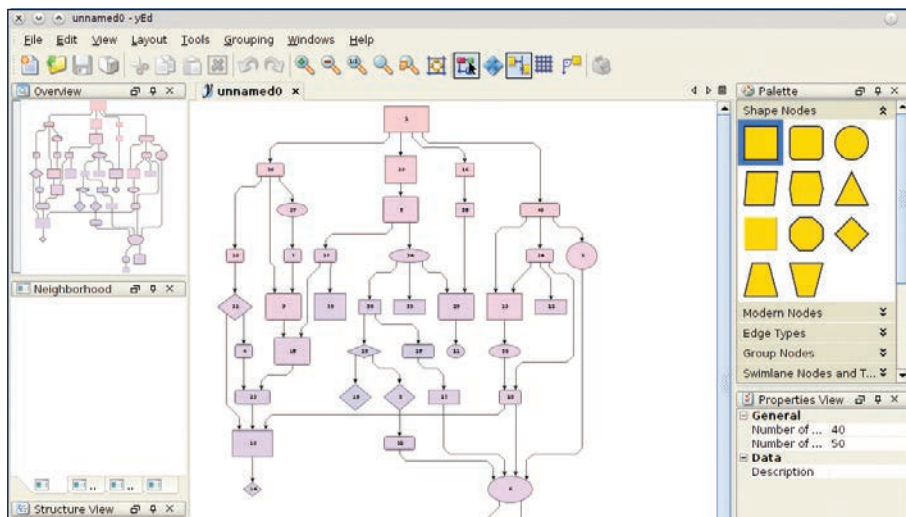
```
java -jar /path/to/yed.jar "$@"
```

из каталога, указанного в переменной `$PATH`, удобно запускает `yEd`, просто набором имени скрипта в командной строке».

Я не очень понял, куда именно поместить файлы программы и скрипт `Bash`?

Джереми Рейнольдс [Jeremy Reynolds]

Установщик делает попытку установить собственную копию Sun JRE. Этот способ очень в стиле Windows, и он может вызывать конфликты с другими программами в системе, или имеет неразрешенные зависимости. На Вашем компьютере уже есть отличная JRE, собранная и поддерживаемая Debian, которая



Установщики, которые пытаются установить все подряд, раздражают. При установке `yEd` пользуйтесь тем, что у вас уже есть.

прекрасно умеет общаться с остальной частью системы, так что «ручная» установка программы все равно будет предпочтительнее. Для хранения программ используются различные каталоги, обычно заканчивающиеся на `bin`. Чтобы просмотреть их, наберите следующую команду:

```
echo $PATH
```

Каталоги `/` и `/usr` трогать не стоит: они предназначены для системных программ. Два варианта — `/usr/local/bin` для глобальной использования и `~/bin` для отдельного пользователя. Если Вы единственный пользователь компьютера, выбор тривиальный. `/usr/local/bin` уже должен быть в переменной `$PATH`, а `~/bin`, скорее всего, там нет. Если Вы решите воспользоваться `~/bin`, создайте этот каталог и добавьте его к своему пути, приписав следующую строку в своем `~/.bash_profile`:

```
export PATH="$PATH:~/bin"
```

Поместите скрипт в созданный каталог и сделайте его исполняемым. Если Вы копируете его в `/usr/local/bin`, это придется делать с правами `root` или воспользовавшись `sudo`. Что касается размещения JAR-файла, то если скрипт сохраняется локально в `~/bin`, то файл JAR удобно сохранить в `~/yEd`. Если скрипт должен быть доступен глобально, то стандартное местоположение для скомпилированных двоичных файлов и файлов Java — `/opt`, так что создайте каталог `/opt/yEd`, поместите туда файл `yed.jar` и внесите соответствующие изменения в скрипт.

Когда скрипт заработает, добавьте с помощью утилиты настройки рабочего стола ярлык на рабочий стол или пункт в меню, чтобы для запуска скрипта не открывать каждый раз терминал.

2 MS-DOS — против Grub?

Недавно я перераспределил разделы на жестком диске, и все закончилось аварийной строкой `Grub` после перезагрузки. Я изменил `fstab` в соответствии с отчетами о разделах в `Gparted` и изменил значения `/dev/sdaX` и `(hd0,msdosX)` в конфигурационном файле загрузки. Но при попытке перезагрузиться мне все равно пришлось устанавливать `prefix` и `root` в правильные значения `(hd0,X)`. Во всех примерах для `Grub` фигурирует, например, `(hd0,msdosX)`.

Так как у меня больше нет раздела `MS` и загрузка происходит из корневого каталога Linux, стоит ли изменить все вхождения `(hd0,msdosX)` в файле настройки на `(hd0,X)`? Это решение кажется очевидным, но я сомневаюсь — вдруг `Grub` как-то связан с `MS-DOS`...

Кен [Ken]

`Grub` ссылается на разделы не так, как Linux, а в `Grub 2` все изменилось еще больше, потому что он поддерживает больше типов разделов. Поскольку `Grub` загружает операционную систему, при его запуске никакая ОС — ни Linux, ни MS-DOS — по определению еще не работает, и загрузчику все равно, что Вы намерены запускать. Ссылка на MS-DOS — это ссылка не на операционную систему, а на таблицу разделов на диске. Последние 30 лет стандартным способом определения разделов на жестком диске является таблица разделов в главной загрузочной записи диска (MBR). Этот способ был популяризован MS-DOS и известен как таблица разделов MS-DOS. Поэтому, когда `Grub` ссылается

на (hd0,msdos1), это означает первый раздел в таблице разделов DOS на hd0. Это можно записать и как (hd0,1) но другой вариант записи предпочтительнее, так как оставляет меньше шансов на путаницу (для Grub, не для Вас). Если Ваш диск был настроен с использованием более новой таблицы разделов GUID (GPT), то первый раздел будет (hd0,gpt1).

Fstab не имеет к этому отношения: он используется ядром Linux для монтирования запускаемых им файловых систем, а ядро на момент запуска Grub еще не запущено. Но поскольку допускается загрузка с изменением параметров в пунктах меню через консоль Grub, исправить разночтение довольно легко. После загрузки можно отредактировать grub.cfg вручную, добавив верные параметры, но Grub 2 может сделать все и сам — для этого запустите команду

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

Чтобы все получилось, возможно, придется изменить пути. В одних дистрибутивах файлы Grub размещаются в /boot/grub, в других — в /boot/grub2; стоит указать неверный путь, и команда не окажет никакого эффекта. В некоторых дистрибутивах команда также может иметь другой синтаксис: grub-mkconfig. Исторически во многих дистрибутивах используется команда grub2, а пути для Grub 2 выбираются так, чтобы они соответствовали старой версии Grub, для упрощения перехода со старой версии на новую.

Если Вы хотите сначала посмотреть, что будет записано в файл, отправьте результат не в grub.cfg, а в программу страничного просмотра.

```
sudo grub2-mkconfig | less
```

Наконец, Grub вовсе не связан с MS-DOS, это программа GNU.

3 Chromium без Flash

В Я обновил Ubuntu с 13.10 на 14.04. В версии 13.10 у меня проигрывалось Flash-видео в браузере Chrome/Chromium. Мне особенно интересны сайты 4OD и BBC iPlayer. В версии 14.04 эти сайты не работают. Сайт 4OD

не грузится, а iPlayer просто показывает вместо программы черный прямоугольник. В Firefox воспроизведение на этих сайтах работает, но я хотел бы решить проблему с Chrome.

Зшли Хит [Ashley Heath]

Каким конкретно браузером Вы пользуетесь, Chrome или Chromium? Это не одно и то же. Chromium — базовый проект с открытым исходным кодом. На его основе был создан браузер Chromium. Chrome также основан на исходниках Chromium, но с рядом изменений, из которых нам важно добавление в браузер кода для воспроизведения Flash. Если Вы пользуетесь Chrome, то для воспроизведения видео через проигрыватель на Flash ничего специально делать не нужно. А если Вы пользуетесь Chromium, проигрыватель необходимо установить.

Эта процедура недавно изменилась из-за внутренних переделок в Chromium — это не имеет отношения к дистрибутиву, только к версии Chromium. Старый пакет adobe-flash больше с Chromium не работает; вместо этого нужно установить pepperflashplugin-nonfree. Он извлекает проигрыватель Flash из Chrome и устанавливает его для использования в Chromium (имя пакета различается в разных дистрибутивах: в некоторых используется chrome-binary-plugins или что-нибудь подобное).

После этого iPlayer от BBC заработает, но 4OD более проблемен. У большинства людей 4OD не работает ни в Chromium, ни в Firefox — даже странно, что Вам удалось заставить его работать в Firefox. Эта проблема вызвана DRM, используемым в 4OD, и ее можно исправить, установив HAL, но этот пакет больше не доступен через репозитории Ubuntu. К счастью, он сохранился в PPA — именно для решения этой проблемы (4OD — не единственный сайт, испытывающий описанные трудности). Установите этот пакет командами

```
sudo add-apt-repository ppa:mjblenner/ppa-hal
sudo apt-get update
```

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 дистрибутивах, 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 применяется команда 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.

```
sudo apt-get install hal
```

Убедитесь, что ни Firefox, ни Chromium не запущены, и удалите каталог ~/.adobe. Некоторые также обнаружили, что после этого нужно переустановить плагин Flash. Еще следует обязательно отключить блокировку любой рекламы с www.channel4.com, поскольку до тех пор, пока Вы не исследуете рекламу на этом канале, Вам не будет позволено просмотреть ни единой программы.



Коротко про...

Компиляция программ

В большинстве дистрибутивов есть репозитории с массой готовых пакетов, но иногда бывает нужно собрать пакет из исходников. Старайтесь при любой возможности пользоваться менеджером пакетов своего дистрибутива, но если придется собрать пакет из исходников, сделать это нетрудно.

```
Распакуйте архив, командой
tar xvf foo-1.2.3.tar.gz
```

Команда tar распознает, как упакован архив. Обычно исходные коды распаковываются

в каталог с именем архива — перейдите в него командой cd foo-1.2.3. Найдите файлы README или Install и прочтите их. Стандартная процедура сборки такова: наберите ./configure, затем make, и закончите su make install или sudo make install.

Первая команда проверяет, что у вас есть необходимые зависимости. Для показа доступных опций наберите ./configure --help.

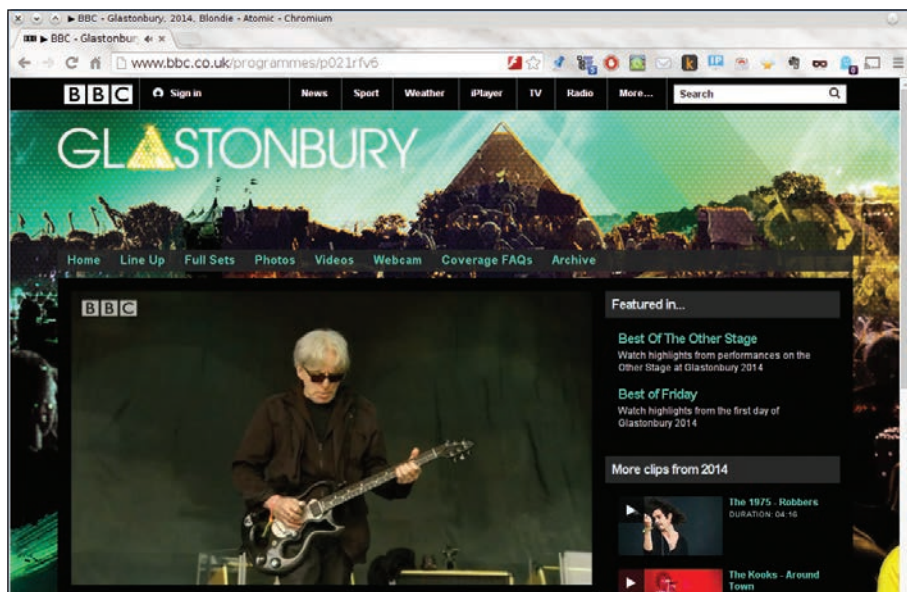
Вторая команда компилирует программу, помещая создаваемые файлы в текущий каталог.

Для компиляции или настройки привилегии root не обязательны, но на третьем этапе скомпилированные файлы копируются в системные каталоги, и вот тут права root уже понадобятся. Поэтому мы запускаем команду через su. В Ubuntu ее нужно заменить на

```
sudo make install
```

Если вы не указали другой путь в параметрах ./configure, скомпилированная программа устанавливается в /usr/local/bin.

Если в системе с форматом пакетов RPM configure жалуется на отсутствие библиотеки там, где библиотека точно имеется, установите дополнительно соответствующий пакет -devel — например, libbar-devel.



► Chrome проигрывает Flash прямо «из коробки». Для Chromium нужно устанавливать отдельный плагин.

4 DVD-молчуны

У меня проблемы с воспроизведением DVD в Ubuntu 12.04, 13.04 и 13.10. Сейчас мои DVD-проигрыватели воспроизводят только один слой DVD-диска. (Я пробовал два: Samsung и iHAS122.) Когда проигрыватель доходит до конца слоя, фильм преждевременно завершается или «падает». Перевернув диск, я могу воспроизвести другую сторону. Я установил *libdvdcss*, которая обычно нужна для воспроизведения DVD. Проблема касается только некоторых DVD, в частности, «Славных парней», но сложно понять, закодированы ли другие фильмы только для первого уровня.

Люк Браунбридж [Luke Brownbridge]

Похоже, Вы путаете двухслойные и двухсторонние диски. Двухсторонний диск — это фактически два диска, прижатые друг к другу, и сегодня их можно встретить не слишком часто. Эти диски нужно переворачивать на середине — что напоминает о пластинках тем, кто достаточно стар, чтобы о них помнить. На двухслойном диске оба слоя расположены на одной стороне диска на немного разной глубине. Поэтому лазер должен перефокусироваться при смене слоев, что часто вызывает небольшую паузу в воспроизведении. В спецификации DVD также описаны двухслойные двухсторонние диски, но такие на практике встречаются редко.

Если у Вас двухслойный диск, то программа может «спотыкаться» об изменение слоя, сочтя прерывание потока данных с диска ошибкой. Вы не указали, какой программой воспроизводите DVD (*libdvdcss* занимается только шифрованием DVD), но для решения проблемы я бы попробовал воспроизвести диск в другой программе. Свои оборудования Вы уже исключили, попробовав два разных привода. Самый всесторонний проигрыватель — пожалуй, *VLC Media Player*, поскольку он справляется почти со всем, что ему ни предложишь. Если Ваши диски не проигрываются

ни в одном из приводов и Ваша проблема ограничена лишь несколькими дисками, возможно, все дело лишь в том, что эти диски повреждены.

5 Незагружаемые DVD

У меня Dell Latitude E4300, и я пытаюсь поставить на него Mint, но он не загружает DVD из привода DVD. Он нехотя пытается раскрутить диск, но потом сдается и переходит к загрузке Windows 7 с жесткого диска. Я пробовал разные диски и дистрибутивы, в том числе *GParted Live*, но безрезультатно. Причем с диска Windows 8 ноутбук загрузился сразу же. Единственный вариант установки, который у меня получился — с флэшки, тут ноутбук загрузился без проблем. Моя последовательность загрузки — USB, DVD и затем жесткий диск. С тех пор я научился загружаться с CD/DVD дисков, нажимая F12 при загрузке и выбирая DVD-привод вручную. Единственный способ, которым мне удается загрузиться — постоянно нажимать F12 после нажатия Enter. Нормально ли это для ноутбуков Dell или есть какая-то проблема с приводом CD/DVD?

Джошуа [Joshua], с форумов

Ноутбук E4300 уже довольно старый, и проблема, похоже, в умирающем приводе оптических дисков — в частности, в лазере. Записываемые диски имеют меньшую отражательную способность по сравнению со штампованными, а у перезаписываемых дисков она и того хуже. Это означает, что привод будет читать штампованные диски, к которым относятся диски LXF (и даже диск с Windows, если Вы настолько отчаянный), а диски, записанные на домашнем компьютере, могут не читаться.

Первыми читаться перестанут перезаписываемые диски, особенно те, которые использовались уже неоднократно. Все, что снижает отражаемость диска, усугубляет ситуацию, поэтому

Помогите нам помочь вам

Ежемесячно мы получаем несколько писем, на которые не в состоянии ответить, поскольку проблема описана в них недостаточно полно. Чтобы дать вам наилучший ответ, нам необходимо 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-файле, который вы сможете приложить к своему письму.

Не уступающий в удобстве альтернативный вариант — *Ishw* (<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
```

шансов у чистого непоцарапанного диска гораздо больше. Постоянное нажатие F12 просто дает приводу больше шансов загрузить загрузочный сектор диска перед тем, как BIOS оставит свои попытки и перейдет к следующему устройству в загрузочной последовательности, т.е. к жесткому диску.

Хотя проблема может быть вызвана неисправным лазером, она также может вызываться грязью на линзе или сочетанием обоих факторов. В этом случае на короткое время может помочь очистка привода с помощью очищающего диска, но долговременный прогноз не очень хороший. На настольном компьютере заменить привод достаточно легко — и дешево, но на ноутбуках это не всегда так.

Как Вы уже обнаружили, копирование ISO на флэшку позволяет обойти проблему, коль скоро у Вас есть доступ к другому компьютеру с надежным DVD-приводом, где диск копируется на флэшку. (Кстати, все DVD LXF можно скопировать на USB.) Загруженные из Интернета ISO-образы, если они не гибридные (т.е. непригодны для записи на флэшку), можно сконвертировать для записи на флэшку программой *isohybrid*, входящей в пакет *SysLinux*. После успешной установки пакета *SysLinux* можно сконвертировать и скопировать ISO следующими командами:



Из-за хрентиллионов вариантов кодеков бывают сложности с проигрыванием видео, но VLC Media Player, похоже, всегда с ними справляется.

```
isohybrid /path/to/iso
dd bs=4k if=/path/to/iso of=/dev/[USBDRIVE]
```

6 Беспроводные причуды

У меня Acer Aspire One 722, на котором была предустановлена Windows 7. Я установил Xubuntu 14.04 в конфигурации с двойной загрузкой. По предыдущему опыту работы с этим ноутбуком и с Ubuntu 12.04, я настроил BIOS так, чтобы нетбук сначала загружался по сети, иначе он зависает, и его приходится перезагружать вручную. Дома на этом ноутбуке у меня хорошо работают как проводная, так и беспроводная сеть. Я пользуюсь Wi-Fi с WPA2 от роутера

и отлично работала в Windows 7? Думаю, дело не в «железе», так как в Windows 7 все работает везде, а скорее в настройках или в проблемах с драйверами — но я сбит с толку! Запуск команды `lspci` дает следующие результаты:

```
ethernet controller: Qualcomm Atheros AR8152 v2.0
fast ethernet (rev c1)
network controller: Broadcom Corp BCM4313
802.11bgn wireless network adapter (rev 01)
```

Роберт Соломон [Robert Solomon]

Приоритетную сетевую загрузку в BIOS выбирать не нужно. Если при более естественном порядке загрузки все еще

ActionТес, который мне предоставил Verizon. Только что я вернулся из поездки, и в моей гостинице был бесплатный Wi-Fi. Мой нетбук видел эту сеть, но не мог получить IP-адрес, когда я щелкал по иконке менеджера сетей. Время ожидания всегда истекало до получения IP-адреса. Однако в Windows 7 я легко подключался к этой точке Wi-Fi.

Почему дома беспроводная сеть работала нормально, а в гостинице — не работала в Xubuntu 14.04

возникают проблемы, стоит проверить обновления BIOS на сайте Acer. Сетевая загрузка как минимум замедлит процесс загрузки на время проверки сетевого хоста для загрузки.

В сетях Wi-Fi — в частности, в некоторых коммерческих и образовательных сетях — используется множество всяких методов аутентификации и шифрования. Сетевые менеджеры не всегда могут правильно определить тип шифрования, хотя в Windows все получилось. Это не удивительно — не могу представить, чтобы в гостинице была беспроводная сеть, которая не работает в Windows. Как Вы отмечаете, с «железом» все в порядке, поэтому либо Network Manager не определяет нужный протокол, либо драйвер BCM4313 в Linux не может обработать этот протокол.

Лучший вариант для Вас (не считая выбора другой гостиницы) — подключиться к Wi-Fi в Windows, зайти в параметры устройства и записать настройки сети. Сделайте скриншот каждой вкладки в свойствах сети и беспроводной сети, чтобы у Вас было все. Загрузитесь обратно в Xubuntu и воспроизведите эти настройки в Network Manager. Это неудобно, но сделать это придется всего один раз для данной гостиницы, а в других гостиницах может использоваться та же схема, так что Вы разом решите все эти проблемы. К сожалению, методов подключения и аутентификации в беспроводных сетях так много, что не все используются очевидными и популярными вариантами. LXF



Часто задаваемые вопросы

Fuse

Полагаю, это не имеет ничего общего с защитой компьютера от неполадок в бытовой электросети [fuse — англ. предохранитель, «пробка»]?

Да, Fuse — это сокращение от “Filesystem in Userspace [Файловая система в пространстве пользователя]”. Это модуль ядра, который позволяет файловым системам работать в пространстве пользователя, а не в ядре.

Так, понятно. Зачем это нужно? Традиционные файловые системы, типа ext4, XFS или ISO9660, являются частью ядра Linux — они либо встроены в само ядро, либо реализованы в виде модулей, подключаемых к ядру во время загрузки. Для этих файловых систем все прекрасно, но тогда код приходится либо включать в основное дерево

ядра, как в случае с этими тремя файловыми системами, либо компилировать для того ядра, которым вы пользуетесь. Стоит изменить ядро, и все внешние модули перестают работать до тех пор, пока вы их не перекомпилируете, о чем хорошо знают обладатели видеокарт Nvidia.

Чем Fuse отличается от них?

Сама Fuse — это модуль ядра, который является частью официального ядра Linux начиная с версии 2.6.14. Данный модуль позволяет связать пространство пользователя, в котором работают файловые системы Fuse, с самим ядром, что позволяет использовать файловые системы, не являющиеся частью ядра Linux.

Все это очень умно, но мне-то оно зачем?

Само по себе почти незачем, если вы сами не хотите программировать файловую систему. Но другие могут это делать, поэтому у нас есть

масса файловых систем Fuse, которых иначе бы не существовало.

Уже есть ext2/3/4, BTRFS, XFS и ReiserFS; нам и вправду нужны новые файловые системы?

Хороший вопрос. Если вы так на это смотрите, то, наверное, нет. Но чаще всего Fuse используется для создания виртуальных файловых систем.

И с реальными-то файловыми системами не вдруг разберешься; что же такое тогда виртуальные?

Виртуальные файловые системы не существуют на диске, но предоставляют информацию в «файлоподобном» интерфейсе вроде /proc. В Fuse их множество: sshfs позволяет монтировать каталоги на удаленном компьютере только через SSH, при этом на удаленном компьютере не должно быть Samba или NFS. EncFS предоставляет зашифрованную файловую систему, которую можно шифровать и монтировать с помощью EncFS. NTFS-3G предоставляет

полный доступ на чтение и запись к разделам NTFS Windows — это одна из нескольких файловых систем Fuse, которая является дисковой файловой системой в традиционном смысле этого слова.

А еще что они умеют делать?

Существуют файловые системы для монтирования различных онлайн-сервисов в виде локальных каталогов. Они позволяют монтировать интернет-сервисы для хранения данных, такие как Amazon S3, или хранилища фотографий, такие как Flickr. Если вашу фотокамеру нельзя подключить как носитель USB и вам приходится загружать фотографии с помощью gPhoto2, то существует файловая система Fuse, которая смонтирует вашу камеру.

Как узнать больше и какие файловые системы еще доступны?

Домашняя страница Fuse доступна по ссылке <http://fuse.sourceforge.net>.



LXF HotPicks



Ричард Смедли

Активист FOSS, Ричард колесит по вязким полям Interweb на своем верном SMED5000 в поисках новых залежей сокровищ открытого кода для вас.

**Kangas Sound Editor QI » CodeWorld » Seafire » Rescatux » Mailpile
» The Music Suite » LFTP » N2048 » Seagull Soup » TeXstudio » Perl**

Звуковые эффекты и создание музыки

Kangas Sound QI

Версия: 4.1.0 Сайт: www.kangasound.com

Kangasound делает программы в сфере музыки и звуковых эффектов. Мы здесь рассмотрим *Kangas Sound Editor QI* — это версия, не зависящая от внешней установки MySQL. Вместо обычной нотной записи она управляет высотой звука через систему соотношения частот. Звуки и инструменты можно создать с нуля, указав гармонические и негармонические призвуки.

Скачайте и установите программу:

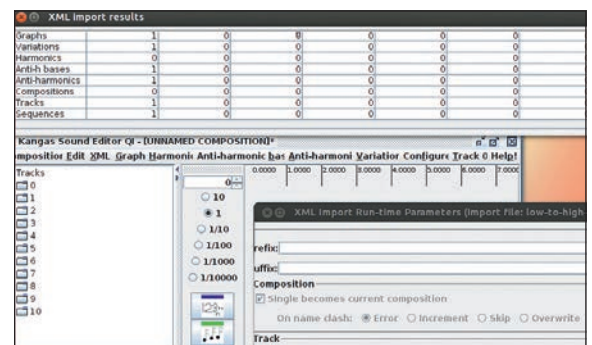
```
java -jar kangas-sound-editor-qi-install-4.1.0.jar
```

Вам придется раз десять нажать ОК, чтобы принять настройки по умолчанию.

Значок в вашем меню для *Kangas Sound Editor QI* — симпатичный мохнатый

кенгуру, или портрет игрушечного. Щелчок на нем откроет переполненный интерфейс, где можно создавать звуки, перетаскивая на панель редактирования поле Sequence с голубыми границами, затем — поле Sound с желтыми границами и поле Harmonic с красными границами. Диалоговые окна предлагают выбор из множества predetermined гармоник.

Это отнюдь не мгновенная процедура, но вы не обязаны работать в одиночку.



И инструменты, и эффекты хранятся как текст в файлах XML, чтобы облегчить сотрудничество.

«Можно создать звуки и инструменты с нуля, указав призвуки.»

Пользователи могут творить коллективно: все, что использует *Kangas Sound Editor QI*, экспортируется и импортируется в виде файла XML. Файлы XML велики, и у старых машин может возникнуть некая натуга при их импорте. Зато это открывает дорогу сторонним инструментам, и можно делать diff по файлам, разделяемым с другими, чтобы увидеть изменения изнутри.

Создав или импортировав звук, нажмите Compute, чтобы создать воспроизводимый звуковой файл. Звуки можно проигрывать по отдельности и настраивать разные аспекты, щелкая по полям разных цветов, которые вы разместили на панели редактирования. Помимо включения Wave Player для воспроизведения звуков, данный релиз добавляет улучшения в просмотр Waveform, Pitch и Amplitude, а также предлагает множество мелких улучшений и исправлений, плюс исправление ошибок.

Мы с удовольствием поиграли в создание звуков, и вы можете сотворить самые замысловатые шумы. Правда, для написания музыки этот инструмент слегка несерьезен по сравнению с основанными на нотах. Но не знающих нотной грамоты, или просто желающих попробовать нечто иное, шумовое искусство порадует.

Исследуем интерфейс Kangas Sound QI

Занято, занято

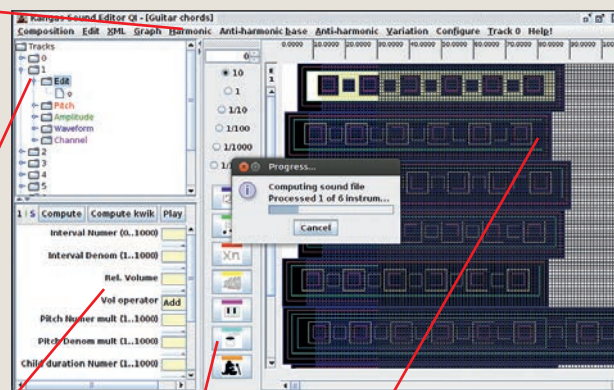
Интерфейс довольно суматошный: *Kangas Sound Editor QI* — мощный инструмент, но придется многому учиться.

Начните отсюда

Track 0 — опорная дорожка, так что начните создание звука с track 1. А вообще-то начать стоит с прочтения онлайн-руководства и просмотра видеоруководства на YouTube.

Выбор звука

Каждый звуковой аспект предлагает несколько вариантов по умолчанию, но оставляет за вами выбор высоты, гармонии, интервала и т.д.



Радужный выбор

Drag-and-drop перемещает нужные вам поля свойств отсюда на панель редактирования.

Сделано

Временная шкала, заполненная звуками, будет выглядеть весьма красочно. Создание звуков с нуля без импорта уже готовых — задача не из легких.

Сетевая среда обучения коду

CodeWorld

Версия: 0.1.0.0 Сайт: <http://bit.ly/CodeWorld>

Переход Криса Смита [Chris Smith] в Google прекратил разработку его курса *CodeWorld* [Мир Кода] в местных школах, но сейчас он изыскал время, чтобы переписывать код. Он уже сделал его доступным для всех желающих обучаться программированию посредством использования Haskell для создания игр и анимации в браузере. Да, мы сказали Haskell.

Установка текущей версии локально включает использование последнего GHC, который работает с GHCJS — это Haskell для компилятора JavaScript. Найдите ответвление GHCJS модифицированной архитектуры [cabal], затем сам GHCJS, и скомандуйте `ghcjs-boot --init`, чтобы установить архитектуру для пары пакетов. Это позволяет скомпилировать *CodeWorld*, но понадобится еще ключ Google API.

Запустив его локально на порте 8080 или разместив на <http://codeworld.info>, вы увидите пустой экран редактирования. Но при нажатии на опции меню внизу

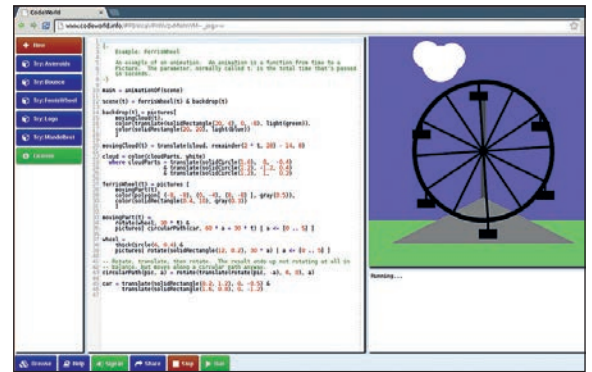
и запуске какого-либо кода он превратится в трехпанельный интерфейс, редактор кода и графический вывод вашей программы.

CodeWorld прячет часть сложностей Haskell. В будущем релизе поведение Haskell по умолчанию изменится, включив нотацию $f(x,y)$, для большего соответствия уровню математики в старшей школе.

Хорошая координация

Среди прочих запланированных изменений в *CodeWorld* — замена холста 500×500 сеткой 20×20 ; координаты x и y варьируются в ней от -10 до 10 . Это сделано, чтобы школьники не использовали пиксели с координатами и обрели лучшее понимание абстрактной плоскости координат.

«Haskell для обучения детей программированию? Это не слишком?»



► **Пробуйте онлайн-версию и ее примеры — но понадобится учетная запись Google, чтобы сохранить свою работу на Google Drive.**

Особо глазастые среди вас могут заметить, что экранный снимок сверху был сделан в *XMonad Window Manager*. Поэтому наш энтузиазм по поводу Haskell не должен никого удивлять — однако... Haskell для обучения детей программированию? Это не слишком?

Смит доказал, что Haskell отлично работает на настоящих уроках в старших классах, и школьники, по склонностям отнюдь не технари, использовали его не менее успешно, чем самопровозглашенные юные программисты. Если вы ищете способ заинтересовать своих подростков программированием, или они уже попробовали Scratch и вы думаете о том, как развивать их дальше, то <http://codeworld.info> — готовое решение.

Облачное хранение и синхронизация

Seafile

Версия: 3.0.4 Сайт: <http://seafile.com>

Мicrosoft сейчас предлагает бесплатно огромные объемы облачного хранения, намекая другим провайдерам удаленного хранения с синхронизацией файлов, что нужно поднять планку. *Seafile* дает вам шансы избежать предательского контроля за вашими файлами — настроив собственное хранилище наподобие *Dropbox* на виртуальном частном сервере (VPS), или в открытом Интернету секторе вашей же сети. *Seafile* — не просто клон *Dropbox*: здесь есть очень удобные функции для совместной работы, определенно оправдывающие занимаемое место на вашем сервере.

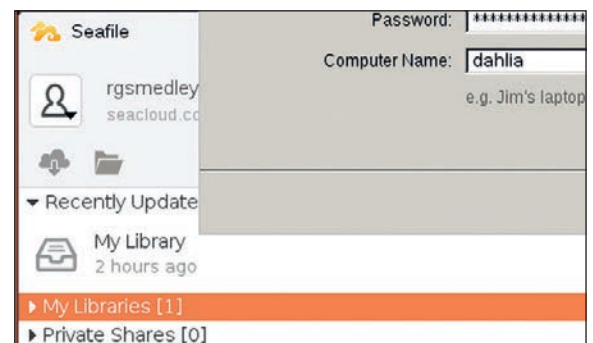
Сервер или облако?

Установка сервера на базе *Django* втягивает ряд зависимостей, но это скорее затратно по времени, чем раздражающе. Если вам сильно не терпится его попробовать, просто скачайте версию для настольного ПК (клиент): она идет с клиентами

и командной строки, и GUI. Соединитесь с <https://seafcloud.cc>; но сначала надо открыть бесплатную учетную запись SeaCloud. Клиент прост в установке — есть даже Deb-пакеты — и есть клиенты для Android и других платформ. Сервер *Seafile* имеет сетевой интерфейс, и вы попадете к своим файлам на сервере или в учетную запись SeaCloud с любой машины.

Как мы отметили, суть *Seafile* не просто в том, чтобы заменить *Dropbox* или Google Drive, но также в обеспечении сотрудничества. Хотя условия <https://seafcloud.cc> отражают заботу о сохранении вашей конфиденциальности — вы можете зашифровать все свои файлы и хранить пароль на клиентах — сама программа

«Суть Seafile не в том, чтобы заменить Dropbox или Google.»



► **Интерфейс клиента Seafile упрощен. Соединиться с серверами Seafile из клиента легко.**

открыта (FOSS). *Seafile* позволяет создавать группы для различных вариантов доступа к каждому документу, встраивать документы в библиотеки с групповым доступом и выполнять передачу WebDAV.

Можно редактировать и комментировать файлы, что, вместе с обменом сообщениями, групповыми обсуждениями и уведомлениями о действиях и версиях файлов, облегчает сотрудничество. Сообщения и уведомления об изменениях можно передать мобильным клиентам.

Обработка конфликтов файлов основана скорее на истории, чем на временной отметке, хотя синхронизацией можно управлять выборочно, через библиотеки или подпапки. Можно синхронизировать много серверов, с контролем версий.

Дистрибутив аварийного восстановления

Rescatux

Версия: 0.32 beta 1 Сайт: <http://bit.ly/Rescatux>

Rescatux (REScue A TUX machine — «спасти машину Тукса») вырос из крошечного, но полезного диска аварийного восстановления Super Grub Rescue. Rescatux добавил 400 МБ полезных пакетов аварийного восстановления на образ CD или USB-брелка, и разработан для 32- или 64-битной архитектуры.

Он загружает диск live Linux — на базе Debian, с аскетичным рабочим столом LXDE — сразу давая выбор типовых сценариев аварийного восстановления, вроде смены пароля или восстановления Grub.

В системе с двойной загрузкой, где Windows затоптал вашу MBR, восстановление Grub — сплошное удовольствие. Раньше всем нам приходилось исправлять подобные проблемы вручную, и обычно выяснялось, что документация не вполне внятная. Печаль усугублялась опасениями, что вы не сможете получить доступ к своей любимой ОС или к своим файлам. Rescatux облегчит ваши страдания благодаря простейшему восстановлению Grub.

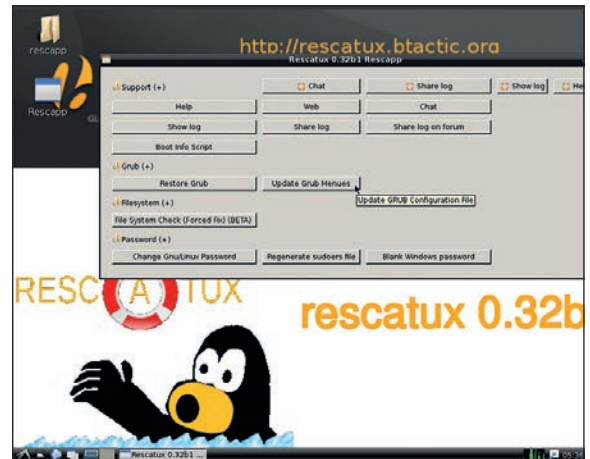
Не щелкайте слишком быстро — обращайтесь с ним, как платежными терминалами, работающими по принципу «не обновлять»: на восстановление некоторых данных у него может уйти не одна секунда.

Готовый чат

Документация на сайте довольно удобная и полезная, а вот wiki на пару релизов отстает. Однако некоторые штрихи, например, кнопка Share Log on Forum, доказывают заботу о том, чтобы облегчить жизнь запаниковавшему пользователю.

Rescatux дружелюбен к новичкам, насколько это возможно для всестороннего инструмента аварийного восстановления; например, в последней версии добавилось исправление файловых систем.

«Rescatux дружелюбен к новичкам, насколько это возможно.»



Обычные сценарии аварийного восстановления диска всего в паре щелчков от протянутой ласты... кхе, руки огорченного пользователя.

Но есть и более продвинутые инструменты для опытных пользователей, в том числе *Gparted*, *extundelete* и *PhotoRec*. Иногда бывает весьма полезен *OS-Uninstaller*.

Включены также браузер и клиент IRC. Последний — весьма продуманное дополнение, потому что у многих пользователей нет доступа к собственным клиентам чата, а огромное собрание тех, кто всегда готов помочь на канале Rescatux, оказывается всего в одном щелчке от вас.

И знаете что? Раньше мы набили оскомину на пингвинов, но сейчас они так редки в приложениях Linux, что снова увидеть еще одного Тукса уже в радость.

Web-почта

Mailpile

Версия: Alpha Сайт: www.mailpile.is

Pешения web-почты существуют в мире открытого кода с 1990-х, и они не стоят на месте; но когда появилась Gmail — поддержанная поисковыми возможностями Google — многие пользователи Интернета, похоже, решили, что лучшего им не найти. *Mailpile* стремится стать не хуже, чем Gmail, но заодно предлагает еще и бонус конфиденциальности, позволяя разместить собственный почтовый сервер, и поддерживает шифрование PGP (pretty good privacy = вполне приличная конфиденциальность).

Разработка пока что на ранней стадии, поэтому корпоративное использование еще не рекомендуется, но полные инструкции по установке находятся в файле *Readme.md*. Клонировать репозиторий GitHub, и скомпилируйте для контейнеров Virtualenv или Docker. Для дистрибутивов на Debian, `sudo make debian-dev` скачает зависимости — сначала с помощью `apt`, затем `pip` и `gems` — и все настроит.

В скрипте установки пропущена зависимость *python-pgpdump*, без установки которой не запускался скрипт запуска *.mp*. Вызов web-интерфейса на localhost переводит вас в собственную командную оболочку *Mailpile*, где можно настраивать программу, выполнять поиск по своей почте и даже составлять сообщения.

Альфа

Хотя web-интерфейс выглядит вполне симпатично, еще предстоит много работы над функциями. Глава проекта Бьярни Рунар Эйнарссон [Bjarni Rúnar Einarsson] и его команда исландских программистов говорят, что *Mailpile* пока не готов к промышленному применению. Мы с ними

«Mailpile быстр, а индексация и присваивание тэгов хороши.»



Привлекательный интерфейс проекта web-почты, который ценит вашу конфиденциальность и безопасность.

согласны, но определенно видим в *Mailpile* большой потенциал.

Mailpile быстр, а индексация и присваивание тэгов работают хорошо, но это не просто попытка стать клоном Gmail. Разработчики *Mailpile*, Pagekite, уверяют, что вы можете использовать *Mailpile* не просто для доступа к Gmail, но также и для шифрования всех ваших сообщений электронной почты, которые там хранятся!

Несмотря на ряд зависимостей, *Mailpile* нагружает сервер весьма незначительно, и вы легко можете использовать его, чтобы разместить свою почту с Raspberry Pi, ноутбука или любого старого VPS (если только сможете установить на него все пакеты). Это клиент электронной почты, за которым стоит приглядывать.

Пакет для создания, обработки и анализа музыки

The Music Suite

Версия: 1.7 Сайт: <http://music-suite.github.io>

The Music Suite — это «семейство специфических языков для создания, обработки или анализа музыки с помощью Haskell», по словам его создателя, Ханса Хёглунда [Hans Höglund]. Ханс не только разработчик, но еще и композитор и дирижер; стоит ли удивляться, что *The Music Suite* предполагает наличие солидных музыкальных способностей — и знания Haskell — чтобы воспользоваться тем, что он умеет делать. Но мы знаем, что читатели **LXF** готовы принять вызов!

Понадобится Haskell в виде Glasgow Haskell Compiler (GHC) и *cabal-install*, оба из вашего менеджера пакетов. Далее, введите

```
cabal update
```

```
cabal install music-suite
```

Что бы вы ни думали обо всех этих репозиториях и программах установки, находящихся вне управления пакетами вашего дистрибутива, они на самом деле облегчают поиск самого свежего ПО во множестве сообществ. Кстати, **cabal** будет скачивать

и распаковывать множество зависимостей, так что вы успеете попить чайку.

При установке зависимости на *music-score-1.7*, *The Music Suite* выдал сбой на одной из машин, по причине “Ambiguous occurrence tabulated [Найден неоднозначный экземпляр в таблице]”. Если с вами произойдет то же, найдите исправленную версию на <https://github.com/armlesshobo/music-score>, скачайте, распакуйте и

```
cabal configure
```

```
cabal install
```

после чего вернитесь к **cabal install music-suite**.

В онлайн-документации масса полезных сведений — настоятельно рекомендуем внимательно прочесть руководство. Чтобы с ходу приступить к делу, возьмите

«Настоятельно рекомендуем внимательно прочесть руководство.»

```
-- LANGUAGE OverloadedStrings, TypeApplications
import Music.Prelude standard hiding (open, play, openAndPlay)
import qualified Music.Score as Score
import Control.Concurrent.Async
import Control.Applicative
import System.Process (system)

-- Anno Part: Cello in Memory of Benjamin Britten (2017)
-- Inspired by the voice transcription
--
main = IO ()
main = open music

ensemble :: [Part]
ensemble = [Solo tubularbells] <+> (divide 2 (tutti violin)) <+> [tutti viola] <+> [tutti cello]

music :: Score Note
music = meta & stretch (2/2) & ("Cello in Memory of Benjamin Britten") (tempo (metronome (1/4) 120))
  where
    meta = id
           . title "Cello in Memory of Benjamin Britten"
           . composer "Arno Part"
           . timeSignature (6/4)
           . tempo (metronome (1/4) 120)

withTintin :: (HasPatches a, Score.Pitch a ~ Behavior Patch) => Pitch -> Score a -> Score a
withTintin p a s => tintin p a s
```

► Мощный и экспрессивный язык Haskell отлично сочетается с мощной и экспрессивной музыкой.

файлы примеров Хёглунда со страниц GitHub и поиграйтесь с кодом, держа под рукой справочник: **music-suite/musicpreludes/tree/master/examples**.

Мы немало рассказывали здесь о форматах хранения музыкальной нотации; *The Music Suite* совершенно спокойно экспортирует MIDI, Lilypond, MusicXML и ABC Notation, а также справляется с экспериментальным импортом нот Sibelius. Поддерживается и импорт MIDI, но учтите предупреждение Хёглунда: *TMS* «может содержать значения времени и высоты нотации; нужны очень умная программа анализа, чтобы конвертировать сырой MIDI в квантованную информацию».

Программа передачи файлов командной строки

LFTP

Версия: 4.5.3 Сайт: <http://lftp.yar.ru>

Учитывая падение спроса на передачу чего-либо по FTP в наши дни, многие из нас в редких случаях использования FTP вместо SCP возятся с жутко устаревшими встроенными клиентами Linux или скачивают *ncftp* из репозитория Ubuntu. Есть и GUI-клиенты, но вам с той же вероятностью понадобится FTP для доступа к области резервного копирования вашего провайдера VPS из безопасной сессии оболочки на ваш сервер.

Только для FTP, *LFTP* уже является дружелюбным к пользователю усовершенствованием, но у него куда больший аппетит к протоколам, чем у большинства клиентов. Он успешно работает с FTPS, HTTP, HTTPS, HFTP, FISH и SFTP.

Мало того. В *LFTP* есть поддержка протокола BitTorrent, включая сидинг, и следующих расширений: Fast, DHT, PEX, Multi-tracker, Metadata и Magnet. *LFTP* поддерживает загрузку из нескольких частей через команду **pgget**:

```
lftp -e 'pget -c -n 5 /path/to/file' sftp://user@futurenet.com
```

и способен зеркально отображать деревья директории любым способом. *LFTP* поддерживает закладки и планирование, что, с нашей точки зрения, весьма удобно.

Следуйте скрипту

Тем, кто в *Bash* как дома, понравится синтаксис команд *LFTP*, похожий на оболочку, который позволяет запускать несколько команд параллельно в фоне и группировать их в (**!s**). Его фоновые способности простираются до непрерывных задач в режиме *pothp*, при условии, что вы закроете X-терминал, в котором его запустили, до того, как он завершит свои задачи.

«Тем, кто в Bash как дома, понравится синтаксис LFTP.»

```
lftp -r -r -r 49H 2012-08-27 02:53 emacs-23.3b.tar.gz
lftp -r -r -r 198 2011-11-26 01:24 emacs-23.3b.tar.gz.sig
lftp -r -r -r 37H 2012-01-28 05:40 emacs-23.4.tar.bz2
lftp -r -r -r 198 2012-01-28 05:40 emacs-23.4.tar.bz2.sig
lftp -r -r -r 46H 2012-01-28 05:28 emacs-23.4.tar.gz
lftp -r -r -r 198 2012-01-28 05:28 emacs-23.4.tar.gz.sig
lftp -r -r -r 40H 2012-06-10 04:25 emacs-24.1.tar.bz2
lftp -r -r -r 130H 2012-06-10 04:25 emacs-24.1.tar.bz2.sig
lftp -r -r -r 49H 2012-06-10 04:10 emacs-24.1.tar.gz
lftp -r -r -r 198 2012-06-10 04:10 emacs-24.1.tar.gz.sig
lftp -r -r -r 40H 2012-08-27 02:19 emacs-24.2.tar.bz2
lftp -r -r -r 198 2012-08-27 02:19 emacs-24.2.tar.bz2.sig
lftp -r -r -r 49H 2012-08-27 02:53 emacs-24.2.tar.gz
lftp -r -r -r 198 2012-08-27 02:53 emacs-24.2.tar.gz.sig
lftp -r -r -r 33H 2012-08-27 02:38 emacs-24.1.tar.xz
lftp -r -r -r 198 2012-08-27 02:38 emacs-24.2.tar.xz.sig
lftp -r -r -r 50H 2013-03-10 22:31 emacs-24.3.tar.gz
lftp -r -r -r 490 2013-03-10 22:31 emacs-24.3.tar.gz.sig
lftp -r -r -r 34H 2013-03-10 22:15 emacs-24.3.tar.xz
lftp -r -r -r 490 2013-03-10 22:15 emacs-24.3.tar.xz.sig
lftp -r -r -r 488K 2001-12-17 14:38 emacs-lisp-intro-2.04.tar.gz
lftp -r -r -r 3.5K 2001-12-17 13:33 emacs-lisp-intro-README-2nd-edition
lftp -r -r -r 3.1H 2005-02-17 07:09 leim-21.4.tar.gz
lftp -r -r -r 65 2005-02-17 07:09 leim-21.4.tar.gz.sig
lftp -r -r -r -- //www.fsf.org/about/dmca-notice
lftp ftp.gnu.org/gnu/emacs> get elisp-manual-21-2.8.tar.gz emacs-lisp-intro-README-2nd-edition |
```

► Возможно, FTP и старый протокол, но *LFTP* делает его эффективным и приносит его функции во множество протоколов передачи.

Основное волшебство, избавляющее от необходимости выполнять скучные повторные задачи, идет из **/etc/lftp.conf**, где можно определить алиасы для серий часто применяемых удаленных команд.

Для настроек, которые незачем применять ко всей системе, имеются команды **~/lftprc** — или **~/lftp/rc**, согласно предпочтениям вашего дистрибутива. Здесь можно не просто задавать опции типа

```
set ftp:use-feat off
```

но также и настраивать *LFTP* для каждого сайта, добавляя доменное имя в указатель после слэша:

```
set ftp:ssl-force on /gnu.org
```

HotGames Развлекательные приложения

Игра с выпадающими фишками

n2048

Версия: 0.1 Сайт: <http://www.dettus.net/n2048>

Вливание более популярных игр — хорошая новость для Linux, хотя дистрибутивы уже и так много лет идут с десятками классических аркад и игр старой школы. Пусть «год настольного Linux» и не наступил, поскольку переход на web-приложения и BYOD сделал настольные ПК не столь актуальными, однако игры никто не отменял.

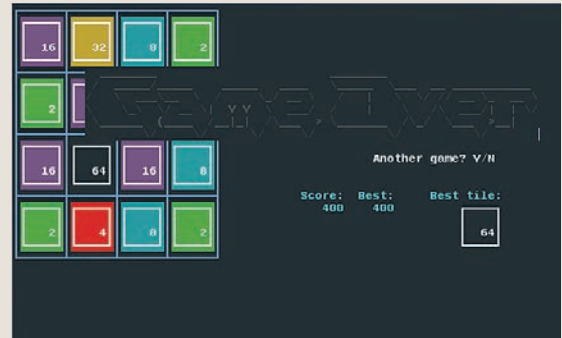
Прелесть открытого кода в том, что каждый может что-то попробовать — прочитать код, увидеть, как он делается, и затем использовать библиотеку с открытым кодом, чтобы написать с ее помощью игру. Типичный высокоуровневый язык для этого — Pygame, тогда как SDL с C++ — для более знающих пользователей. Чтобы подлинно погрузиться в самое сердце среды *nix, напишите собственную игру для командной строки на C, используя библиотеку *ncurses*.

Однако автор *n2048*, Томас Деттбарн [Thomas Dettbarn], вовсе не новичок в командной строке. Помимо шестнадцатеричного редактора *DHEX* (очень удобного с *diff*), он написал и другие низкоуровневые инструменты — например, *nMicrocoder*: это инструмент EDA *Ncurses* для создания микрокода. Что и объясняет элегантность кода, нетипичную для релиза 0.1.

Веселье — сразу

Игровой процесс прост: заставляя фишки соскальзывать покачиванием доски. Быстрое перемещение в случайных

«Периодически вы проигрываете. И начинаете все заново.»



► Вы-то думали, что любите отстреливать зомби, но вдруг ощущаете странный позыв гонять фишки по доске.

направлениях кажется более эффективной тактикой, чем любая другая — возможно, в следующих версиях это будет изменено?

Когда вы качаете доску с помощью кнопок со стрелками — или традиционных альтернатив WASD либо HJKL — фишки соскальзывают вместе, и это повышает их ценность. Вы должны достичь сказочной фишки 2048. По каким-то причинам вы вдруг обнаружите, что перемещаете фишки все быстрее и быстрее... возможно, потому что игра не тормозит, и вы просто-напросто можете это делать. Периодически вы проигрываете. И начинаете все заново.

Стрелялка

Seagull soup

Версия: 1.0.5 Сайт: <http://seagullsoup.sf.net>

Говорят, что аркада оказывает терапевтический эффект. Если в вашей памяти свежи воспоминания о пикнике, загубленном налетом агрессивных чаек, *Seagull Soup* явно сэкономит вам на консультациях психолога, позволив выместить злобу на этих виртуальных летунах в умилительно-дурацкой стрелялке.

Игра написана на C с библиотеками SDL, и проста в установке. Скачайте и распакуйте ZIP файл и запустите *make*. Вы получите *seagullsoup.exe* объемом 74 КБ, библиотеки SDL и 8 МБ файлов с графикой. ZIP также содержит исполняемый файл Windows, не совсем в русле прочей информации. К счастью, исходник вполне читаем, и на него стоит взглянуть, если вы затеваете попробовать свой первый бит игрового кода C и SDL. Сама по себе игра проста. По сути, это невинный способ весело скоротать пару

свободных минут, даже если вы ничего не имеете против чаек.

Сложность игры повышается по мере прохождения уровней — птицы совершают всякие движения, имеют разное оружие и разную сопротивляемость вашим выстрелам. Настает ночь, и увеличивается число птиц и скорость их полета. Хотя управлять можно с клавиатуры (стрелки для перемещения, пробел для стрельбы и Ctrl для самонаводящихся ракет), все же на самом деле она написана для джойстика, в классическом аркадном стиле.

Когда-нибудь у вас да наступит момент славы, но в любом случае вы дадите

«В любом случае, вы дадите урок этим гадким чайкам.»



► Однако это очень даже неплохо — отстреливать сотни виртуальных чаек в небе...

урок этим гадким чайкам. Не таскать им больше у вас бутерброды с сыром!

Автор игры, Роберт Стенстрём [Robert Stenström], написал еще две вариации на эту тему, так что у него явно непростые отношения с чайками. Может быть, в детстве его запугали пьесой Чехова?

TeXstudio

Версия: 2.8.0 Сайт: <http://texstudio.sf.net>

Latex и TeX — до мозга костей инструменты командной строки; как любой другой код, их можно написать и отредактировать в редакторе с GUI. В случае с LaTeX это отнюдь не излишняя роскошь — немного наберется возможностей изучить LaTeX без тщательной проработки всей документации инструмента.

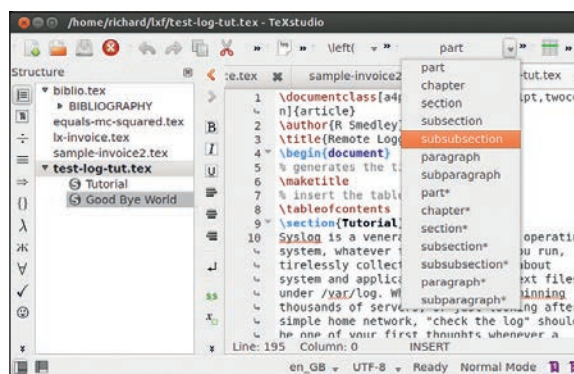
Для трудолюбивых и пытливых, ищущих идеальный способ ввода, это рука помощи. TeXstudio выделяет ваши документы, потом ищет и заменяет текст, проверяет правописание и грамматику и предлагает предпросмотр готового документа в виде PDF.

Настраиваемая панель инструментов и прочие части GUI в TeXstudio полностью нацелены на максимальное упрощение без ущерба для обучения. Поскольку в результате операций GUI вы видите изменения в документах TeX, получается готовый ответ на вопросы типа «Как это сделать в LaTeX?». Древоидная структура документа, автозаполнение и встроенная

помощь тоже весьма удобны. Опытные пользователи LaTeX чрезвычайно уважают TeXstudio.

На сайте проекта TeXstudio имеются пакеты для новых и более старых версий Arch, CentOS, Debian, Fedora, OpenSUSE и Ubuntu, но также и для FreeBSD, Mac, и OS/2, что определенно доказывает глубокую преданность и заинтересованность пользователей TeXstudio.

Хотя для LaTeX и не нужен GUI, он определенно помогает, пока вы только еще осваиваетесь.



Язык скриптов (и цепная пила швейцарской армии)

Perl

Версия: 5.20 Сайт: www.perl.org

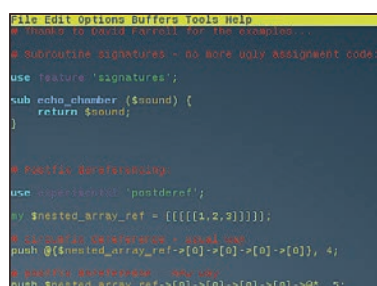
Набитый превосходными новыми функциями, от подписей подпрограмм (экспериментальная функция) до разыменования Postfix, Perl 5.20 показывает, что в Perl 5 осталось много жизни и простора для разработки даже спустя 14 лет после объявления о выходе языка программирования Perl 6.

В эту новую версию Perl вложено много труда: в прошлом году — «470 000 строк изменений в 2900 файлах от 124 авторов». В основном это изменения эволюционные, а не революционные, и включают мелкие поправки в системе производительности и безопасности, а также долгожданные отладки совместимости.

Изменения в документации и диагностике показывают, что дружелюбие Perl к пользователю, вероятно, недооценивается. У Perl имеется огромное сообщество, которое вы откроете для себя, прочтя уведомление о релизе, где вы найдете среди прочего некрологи всем известным

программистам Perl, умершим с момента последнего релиза.

Обновление предлагает повышенную 64-битную производительность, а также некоторые улучшения и исправления в работе с регулярными выражениями, которая является истинным алмазом в короне Perl. Самым примечательным для Linux является тот факт, что Perl 5.20 компилируется на Android. Только подумайте, сколько готовых старых скриптов вы сможете запустить на своем планшете! **LXF**



Современный Perl по-прежнему хорош для связывания кода и «строк шума», но и обзавелся массой передовых возможностей.

Также вышли

Новые и обновленные программы, тоже достойные внимания...

Crane

ORM для Common Lisp, который не пьет ORM Kool Aid.
<http://eudoxia0.github.io/crane>



ORM Crane способен выдать вам и простоту, и мощь, по вашему запросу.

Members Area

Удобен для британских Maker/Hackerspaces, или им подобных некоммерческих групп.
<https://github.com/members-area>

mbank-cli 1.0

Интерфейс командной строки для банковской онлайн-системы mBank.
<https://code.google.com/p/mbank-cli/>

Possum 0.5.1

ПО для кассовых машин (POS) ресторанов, баров и пивных.
<http://possum.readthedocs.org/en/latest>

decotengu 0.13.0

Библиотека декомпрессии в дайвинге, использующая модель декомпрессии Бульманна [Buhlmann].
<http://wrobell.it-zone.org/decotengu>

Moka 3.0

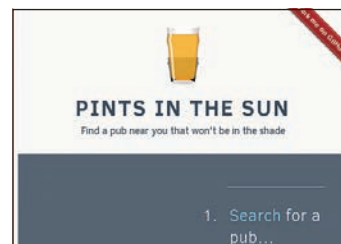
Проект элегантного дизайна для значков, тем GTK и прочих элементов брендинга.
<http://moka-project.com>

Privacy Badger

Это дополнение к браузеру не позволит третьим сторонам отслеживать ваше пребывание на сайтах.
www.eff.org/privacybadger

Pints in the Sun

Программа определения места, основанная на OpenStreetMap, чтобы найти слабое местечко для вечерней выпивки.
<http://pintsinthesun.co.uk>



Посидите на солнышке — с пивом и программами с открытым кодом.

На диске

Дистрибутивы, приложения, игры, подкасты и всякое-разное...

Лучшее из Интернета, упакованное в 8 ГБ качественного DVD.



Миленькая шляпка

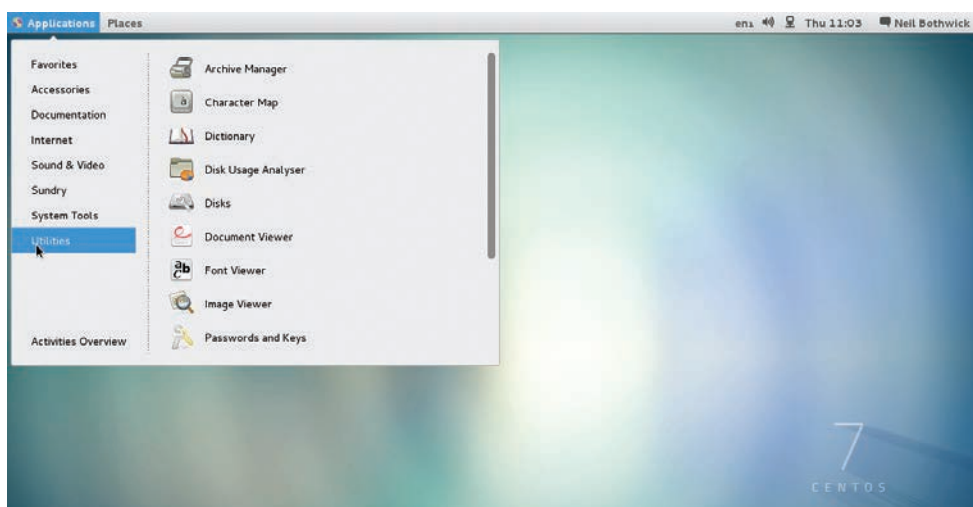
Один из дистрибутивов на диске этого месяца — символ взаимоотношений между открытым кодом и большим бизнесом. Red Hat — это и крупнейшая компания Linux, и самый плодотворный вкладчик кода в ядро Linux, используемое также и его конкурентами. Банальная мудрость предположила бы, что лить воду на мельницу конкурента — во вред бизнесу Red Hat, однако выходит, что это не так. Да, они отвечают за 12% кода в ядре, но взамен бесплатно получают другие 88% — вот вам сделка в духе открытого кода.

Открывать весь свой код в Red Hat не обязаны, но они это делают. Red Hat не только позволяет таким дистрибутивам, как CentOS и Scientific Linux, производить не брендовую, но во всем остальном совершенно идентичную версию RHEL: они также активно поощряют и частично финансируют CentOS, предоставляя работу некоторым из его разработчиков. И не только в области кода ядра: Red Hat принимают на работу также разработчиков таких программ, как *PulseAudio* и *Systemd*, программы, которые все больше применяются другими дистрибутивами. Они желают захватить весь мир? Возможно; но даже если и так, то в очень милой форме.

Neil

Дистрибутив Linux

CentOS 7.0



Один дистрибутив обсуждают относительно мало, хотя он работает на множестве компьютеров Linux по всему миру. Это Red Hat Enterprise Linux (RHEL) — дистрибутив, одновременно и полностью коммерческий, и с открытым кодом. RHEL не распространяется свободно, как многие другие дистрибутивы, зато его исходник выложен

в открытый доступ, а значит, на нем допускается строить другие дистрибутивы. CentOS — один из таких, и это практически RHEL, разве что без брендинга Red Hat. Если вы хотите попробовать дистрибутив, прочный и стабильный, как RHEL, не оплачивая при этом контракта Red Hat, то CentOS 7.0 более чем достоин вашего внимания.

Дистрибутив Linux

Peppermint Five

У традиционных рабочих столов вроде Gnome и KDE есть тенденция накапливать функции и наращивать требования к вашему оборудованию. И рабочие столы в этом не одиноки. Конечно, есть еще тенденция перехода от локальной работы к облачным технологиям. Peppermint — это дистрибутив Linux, отвечающий обеим этим тенденциям. Он настроен для работы в облаке, но не до такой степени, как Chrome OS, которая в отсутствие интернет-соединения весьма ограничена. Здесь имеются все инструменты и традиционный рабочий стол, очень легковесный. Основан он на Ubuntu, но с другим менеджером окон и собственным инструментарием. Как и подсказывает его название, нам кажется, что Peppermint [англ. «перечная мята»] — свежее веяние.

Rescue CD — полезные инструменты, но часто требуют определенных знаний, которые вы не сможете получить без соединения с Интернетом. Rescatux — спасательный CD с дружелюбным интерфейсом для стандартных задач восстановления системы, таких, как отладка загрузчика *Grub*. Держите его под рукой на случай, если все пойдет наперекосяк.

» Важно ВНИМАНИЕ!

Порченные диски

В маловероятном случае какого-то дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу disks@linuxformat.ru или телефону +7 (812) 309-0686.





Новичок в Linux? Начните отсюда!

- » Что такое Linux? Как его установить?
- » Есть ли в нем эквивалент *MS Office*?
- » Зачем нужна командная строка?
- » Как устанавливать программы?

Ответы приводятся в [Index.html](#) на диске.

Из журнала

Разгоняем Linux

Вдарьте по газам вашего компьютера.

Коды руководств

Программы и примеры для учебников этого месяца.

Сравнение

Управляйте своим ПК голосовыми командами.

LXFHotPicks

CodeWorld

Попробуйте среду обучения на базе браузера, которая использует Haskell для создания игр и анимации.

Kangas Sound Qi

Создавайте с нуля звуки и множество инструментов в этом звуковом комплекте.

LFTP

Клиент FTP, который поддерживает множество протоколов передачи, в том числе BitTorrent и сидинг.

Mailpile

Используйте шифрование PGP для большей конфиденциальности электронной web-почты.

n2048

Усовершенствованная головоломка с фишками от создателя шестнадцатеричного редактора DHEX.

Perl 5.20

Теперь с улучшенной 64-битной производительностью и способностью компилироваться даже на Android.

Rescatux

С помощью удобных инструментов этого live-дистрибутива, например, для восстановления Grub, чините свой ПК.

Seafile

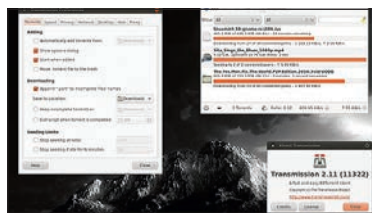
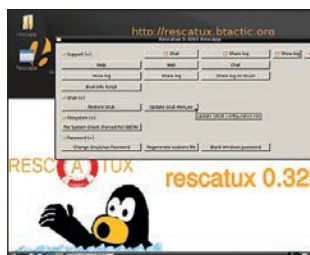
Настройте личное хранилище в облаке наподобие Dropbox.

TeXStudio

Рука помощи начинающим пользователям LaTeX, в виде редактора выделения синтаксиса.

The Music Suite

Создавайте, обрабатывайте и анализируйте музыку, используя Haskell. Для любящих испытания читателей LXF.



И еще!

Системные инструменты

Главное

Checkinstall Установка tar-архива с помощью менеджера пакетов.

GNU Core Utils Основные утилиты, обязанные быть в каждой операционной системе.

Hardinfo Инструмент для тестирования системы.

Ядро Исходник последней версии ядра.

Memtest86+ Проверьте свое ОЗУ на сбои.

Plop Простой менеджер загрузки для запуска операционных систем.

RaWrite Создавайте загрузочные диски в Windows.

SBM Независимый от ОС менеджер загрузки с простым в работе интерфейсом.

WvDial Соединяйтесь с Интернетом через телефонный модем.

Чтение

Книжная полка

The Cathedral and the Bazaar [Собор и Базар] Классический текст Эрика С. Реймонда [Eric S Raymond], объясняющий преимущества открытой разработки.

Linux Kernel in a Nutshell [Ядро Linux в двух словах] Введение в ядро, написанное магистром хакеров Греггом Кроа-Хартманом [Greg Kroah-Hartman].

Справочник администратора Debian Руководство для системных администраторов.

Словарь Linux Linux от A до Z.

Окунитесь в Python Мастер-класс по этому популярному языку.

Руководство по созданию скриптов Bourne Shell Осваивайте скрипты оболочки.

Расширенное руководство по созданию скриптов Bash Больше информации для опытных пользователей.

Intro to Linux Руководство для новичков.

Руководство сисадмина Основы работы с системой.

Обзор утилит Обзор инструментария GNU.

Подкасты



Сезон 6 Эпизод 3

Все еще эпизод 2. Нам жаль, но мы были жутко заняты, из-за всех сил делаем для вас отличный журнал, вместо болтовни на темы, которые и так есть в любой приличной ленте RSS.

Новости Red Hat (все еще) объединяет силы с CentOS, Ye Olde SteamOS (все еще) сияет, Steam Machines (все еще) не столь сияют, почему Steam VR (все еще) пугающе хорошеет, взломанные холодильники (все еще) шлют вам спам; и дата (все еще) ожидаемого Конца Света для релизов Ubuntu.

Открытия недели Дистрибутивы для слабовидящих, общий доступ к данным NHS — хорошо или плохо? Изучение жестких дисков BlackBlaze со странным итогом.



Пропустили номер?



Закажите его через сайт www.linuxformat.ru в «ГНУ/Линуксцентре»! Журналы доставляются и в печатной, и в электронной форме, так что с момента открытия браузера до получения нужного вам выпуска LXF может пройти всего пара минут!

Прямо сейчас для заказа доступны следующие номера:



LXF185
Июль 2014

250 руб.

- » **Смастерим компьютер** Это проще, чем вы думали
- » **Сам себе Википедия** Выбираем движок wiki
- » **UEFI** Отправим BIOS на покой
- » **Minecraft Pi** Игры в блочной вселенной

LXFDVD: Ubuntu 14.04 LTS, MX-14, Trisquel 60.1, Linux Mint 17, Grml 2014.03, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_185/
PDF-версия: shop.linuxformat.ru/elxf_185/



LXF186
Август 2014

250 руб.

- » **Взломаем Ubuntu** Пусть пляшет под нашу дудку
- » **Экономные дистрибутивы** К чему ненужные роскоши?
- » **GoboLinux** Нет, это не про гоблинов
- » **Дни Денди и Sega** Их удалось вернуть

LXFDVD: Crunchbang 11, Elive 2.2.2 beta, Manjaro OpenBox 0.8.9, Puppy Slacko 5.7, Calculate Linux 13.19, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_186/
PDF-версия: shop.linuxformat.ru/elxf_186/



LXF187
Сентябрь 2014

250 руб.

- » **Уходим с Windows** Подстелим соломки для прыжка
- » **Музыка, туш!** Лучшие медиа-проигрыватели
- » **OIN** Пресечь патентных троллей
- » **Wine** — не эмулятор А тогда что?

LXFDVD: Mint 17, Ubuntu 14.04 LTS, SolydX 201405, CAINE 5.0, PCLinuxOS 2014.08, SystemRescueCD, Gparted, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_187/
PDF-версия: shop.linuxformat.ru/elxf_187/

А чтобы не упустить ничего, оформите подписку! Все, кто подписался на печатную версию журнала через www.linuxformat.ru/subscribe/ или www.linuxcenter.ru, получают электронную версию в подарок! На сайте shop.linuxformat.ru вы также сможете приобрести предыдущие выпуски LXF.

Подписывайтесь на сайте www.linuxformat.ru/subscribe

Телефоны отдела подписки:

- » Санкт-Петербург (812) 309-0686
- » Москва (499) 271-4954



Linux Format ВКонтакте

Вступайте в нашу
группу vk.com/linuxform

На странице LXF ВКонтакте вы найдете:

- » Новости о Linux
- » Статьи из архива LXF
- » Анонс свежего выпуска LXF и часть статей из него
- » Живое общение и онлайн-консультацию по подписке на наше издание

Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials [Главное]» на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя_программы-1.0.1.i386.rpm** — вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя_программы-1.0.1.i386.deb** — такой же пакет, но уже для Debian;
- » **имя_программы-1.0.1.tar.gz** — обычно это исходный код;
- » **имя_программы-1.0.1.tgz** — тот же файл, что и выше этажом по списку: «tgz» — это сокращение от «tar.gz»;
- » **имя_программы-1.0.1.tar.bz2** — тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** — также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** — двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** — двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** — версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Внимательно прочтите это перед тем, как использовать LXF DVD!



Октябрь 2014
LXF DVD 188

LINUX
ФОРМАТ

Октябрь 2014
LXF DVD 188

LINUX
ФОРМАТ

Содержание

LINUX FORMAT

Страница 1

ДИСТРИБУТИВЫ
Arch Linux 2014.7 (32- и 64-разрядная сборка)
CentOS 7.0 (64-разрядная сборка)
Netrunner 14 (64-разрядная сборка)
Repremint Five (64-разрядная сборка)
Rescatux 0.32 (32- и 64-разрядная сборка)

ПРОГРАММЫ РАСПОЗНАВАНИЯ

РЕЧИ
FreeSpeech
Palaver
Simon
Speech-App

NOTPICKS

CodelWorld Сетевая среда обучения коду
Kangas Sound QI Звуковые эффекты и создание музыки

LTPR Программа командной строки для передачи файлов
Mailpile Веб-почта
n2048 Игра с выпадающими фишками
Perl 5.20 Язык скриптов
Rescatux Дистрибутив аварийного восстановления
Seafile Облачное хранение и синхронизация
Seagull Soup Стрелялка
TeXStudio Редактор выделения синтаксиса LaTeX
The Music Suite Пакет для создания, обработки и анализа музыки

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

Страница 2

ДИСТРИБУТИВЫ
Knoppix 7.4.1 Установочный + Live (32- и 64-разрядная сборка)
Clonezilla 2.2.4-12 Live (ISO-образ, 32-разрядная сборка)

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 — Греггом 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

Пожалуйста, перед использованием Аджунто Диска ознакомьтесь с инструкцией, опубликованной в журнале на стр. 109!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДВОЕКЛЕТНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех этапах производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аналитический сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD. Предоставляя этот DVD, редакция Linux Format гарантирует только качество печати, которое подтверждено в соответствии с требованиями ISO 9001. Мы не несем ответственности за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD. Предоставляя этот DVD, редакция Linux Format гарантирует только качество печати, которое подтверждено в соответствии с требованиями ISO 9001.

Тираж изготовлен ООО «Уральский электронный завод», 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПТР ВАР 77-15.

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу — это обратиться к программе *cdrecord*. Для всех перечисленных ниже действий потребуются права суперпользователя-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 (скорее всего, в вашей системе присутствует только 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
```

Для записи ISO-образа вам осталось набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не принадлежите к любителям командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее от имени 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 нажмите на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Использовать Linux для записи компакт-диска не обязательно. Программы вроде *cdrecord* просто переносят двоичные данные на чистую матрицу. Все необходимые файлы уже включены в ISO-образ, который распознается любой операционной системой, будь то Linux, Windows, Mac OS X или AmigaOS.

Нет устройства для записи дисков?

Если у вас нет устройства, с помощью которого можно было бы записать образ на диск, можно найти какого-нибудь друга или организацию, у кого есть компьютер с дисководом, и прожечь диск у них. Опять-таки, вам 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.



UNIXFORUM.org



Есть вопросы?
Задавайте!

<http://unixforum.org/>

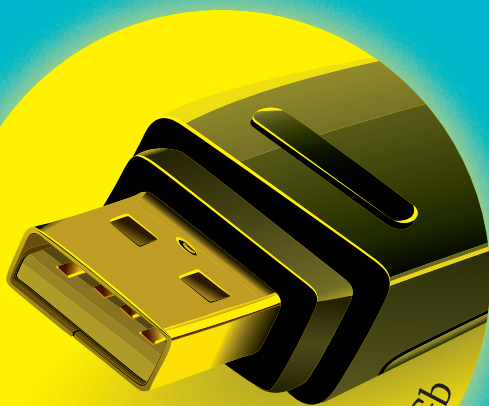
Отдел дистрибьюции ГНУ/Линуксцентра приглашает дилеров и дистрибьюторов к сотрудничеству!

Широкая сеть представительств
в разных городах России
позволит вам оптимизировать
процессы логистики и доставки товара

ПОДРОБНЕЕ О ПАРТНЕРСКОЙ ПРОГРАММЕ:
WWW.LINUXCENTER.RU/PARTNER/



ДИСТРИБУТИВЫ GNU/LINUX НА USB FLASH



Linux Mint 16



Ubuntu 14.04

А ТАКЖЕ
версии для юриди-
ческих лиц —
с лицензионным
договором
присоединения

USB Flash 8 Gb

495 рублей

www.linuxcenter.ru/shop/distros/usb/



UnixEducationCenter

ОБУЧЕНИЕ И СЕРТИФИКАЦИЯ ОС LINUX



ПО ВСЕМ ВОЗМОЖНЫМ УРОВНЯМ:

RHCSA – RED HAT CERTIFIED SYSTEM ADMINISTRATOR

RHCE – RED HAT CERTIFIED ENGINEER

RHCA – RED HAT CERTIFIED ARCHITECT

www.unixedu.ru

+7 (812) 611-15-75 mail@unixedu.ru

Санкт-Петербург, Черноморский пер., д.4

ГНУ/Линуксцентр
приглашает на работу!



ВАКАНСИЯ: Разработчик систем электронного документооборота

ТРЕБОВАНИЯ:

- » Опыт внедрения и сопровождения систем электронного документооборота на базе Alfresco
- » Умение описать бизнес-процесс
- » Знание SQL, понимание принципов построения и функционирования баз данных
- » Знание СЭД, отличных от Alfresco, приветствуется

ОБЯЗАННОСТИ:

- » Разработка систем с нуля, сопровождение
- » Настройка системы в соответствии с требованиями бизнеса
- » Написание инструкций для конечных пользователей

ПОДРОБНЕЕ: www.linuxcenter.ru/vacancy/

Футболки GNU/Linux

БОЛЕЕ 100 ВИДОВ. ВСЕ РАЗМЕРЫ И ЦВЕТА



WWW.LINUXCENTER.RU/SHOP/GIFTS/

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. Выпуск издания осуществлен при финансовой поддержке Федерального агентства по печати и массовым коммуникациям. 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

Александр Баракин

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Андрей Смирнов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатель

ЗАО «Мезон.Ру»

Отпечатано в типографии ООО «ЛД-ПРИНТ»

196644, Санкт-Петербург, Колпинский р-н, пос. Саперный, территория предприятия «Балтика», д. б/н, лит. Ф
Тел. (812) 462-8383, e-mail: office@ldprint.ru
Заказ 11412

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор [Neil Mohr] neil.mohr@futurenet.com

Заместитель редактора Мэтью Хэнсон [Matthew Hanson]

matthew.hanson@futurenet.com

Научный редактор Джонни Бидвелл [Jonni Bidwell] jonni.bidwell@futurenet.com

Выпускающий редактор Крис Торнетт [Chris Thornett]

chris.thornett@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоза

[Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Джонни Бидвелл [Jonni Bidwell], Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Джолион Браун [Jolyon Brown], Кент Ельчук [Kent Elchuk], Марко Фиоретти [Marco Fioretti], Мэтт Хэнсон [Matt Hanson], Дэвид Хейвард [David Hayward], Джульетта Кемп [Juliet Kemp], Эндрю Моллетт [Andrew Mallett], Нейл Мор [Neil Mohr], Лес Паундер [Les Pounder], Маянк Шарма [Mayank Sharma], Ричард Смедли [Richard Smedley], Евгений Балдин, Дмитрий Кузнецов, Михаил Остапкевич, Алексей Федорчук

Ассистенты по выпуску Элис Липскомб-Саутвелл [Alice Lipscombe-Southwell], Эстер Ньюмен [Esther Newman], Алекс Саммерсби [Alex Summerby]

Художественный ассистент Сара Кларк [Sarah Clark]

Иллюстрации Шейн Коллиндж [Shane Collinge], Саймон Миддлвек [Simon Middleweek], iStock Photo

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Тел. +44 01225 442244, email: linuxformat@futurenet.com

РОССИЯ:

Санкт-Петербург (редакция):

пр. Медиков, 5, корп. 7. Тел. +7 (812) 309-0686

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 вам 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 диски — CD или DVD, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственность за повреждение или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds]. "GNU/Linux" заменяется на "Linux" в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futureplc.com>.

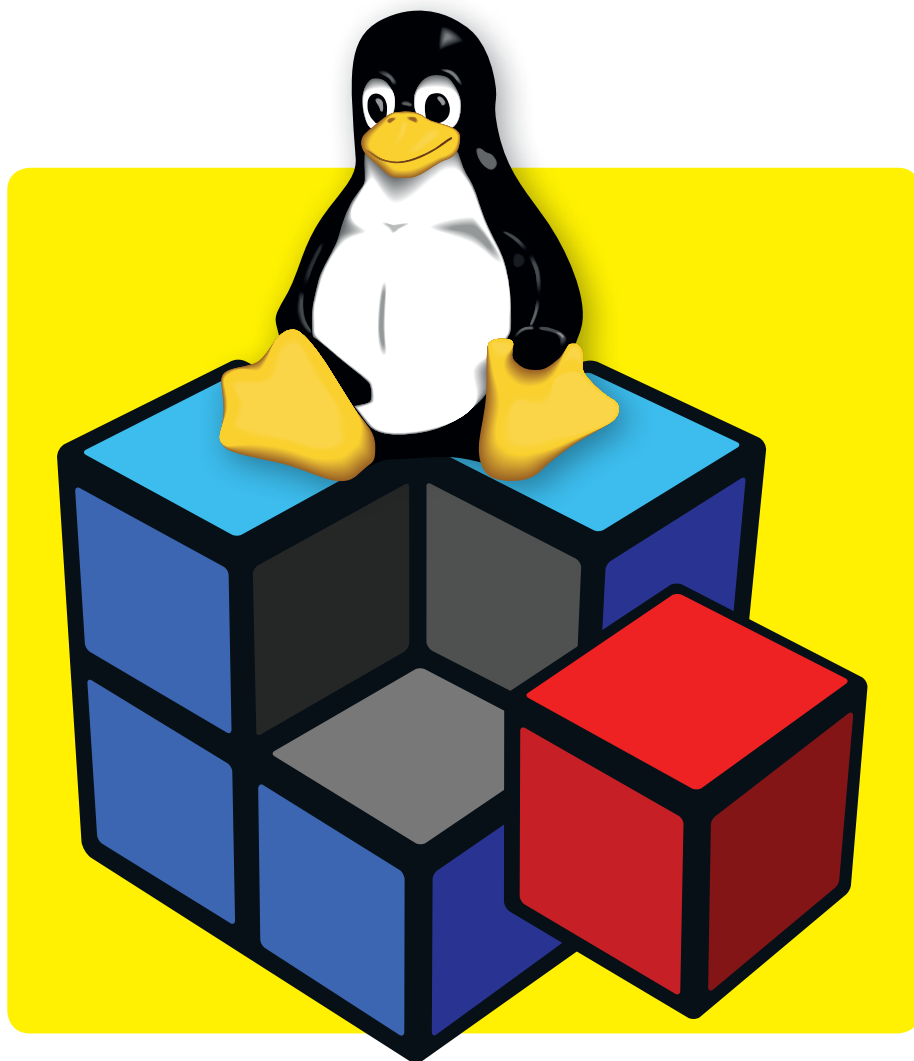


© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

16+



В ноябрьском номере

Решим любую проблему Linux

Доколе?! Довольно! Вот вам супер-руководство по пресечению всех ваших бед в Linux. Учимся диагностировать и истреблять наиболее общие проблемы.

Обзор Chromebook'ов!

Проводим тест-драйв лучших моделей.

Постквантовое шифрование

Ближе к природе: шифрование нового поколения.

Linux на Android'e

Запускать Linux на устройствах с Android — да легко.

Содержание будущих выпусков может меняться — вдруг у нас будут проблемы с установкой драйвера принтера...

Новое поколение средств защиты

Межсетевые экраны ССПТ, не имеющие IP-адреса

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

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

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

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

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

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

Дополнительное устройство защиты для:

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

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

HETZNER DEDICATED SERVERS

HETZNER
ONLINE

В ЦЕЛОСТИ И СОХРАННОСТИ!

ХРАНЕНИЕ ДАННЫХ В ГЕРМАНИИ



ВЫДЕЛЕННЫЙ СЕРВЕР PX60

- Intel®Xeon® E3-1270 v3 Quadcore Haswell
- 32 ГБ ECC RAM
- 2 x 2 ТБ 6 Гбит/с SATA 7200 об/м Enterprise класс Software-RAID 1
- 20 ТБ Трафик*
- Операционная система Linux
- Подключение 1 Гбит порт
- Гарантировано 200 Мбит/с
- IPv6 подсеть (/64)
- Без минимального контракта

2600

рублей в месяц
+ Установка 3700 рублей



ВЫДЕЛЕННЫЙ СЕРВЕР PX70-SSD

- Intel®Xeon® E3-1270 v3 Quadcore Haswell
- 32 ГБ ECC RAM
- 2 x 480 ГБ 6 Гбит/с SATA SSD Data Center Series
- 30 ТБ Трафик*
- Операционная система Linux
- Подключение 1 Гбит порт
- Гарантировано 200 Мбит/с
- IPv6 подсеть (/64)
- Без минимального контракта

3700

рублей в месяц
+ Установка 3700 рублей



Безопасное защищённое хранение ваших данных в собственных дата-центрах немецкой компании Hetzner Online, соответствующее высоким требованиям к стандартам защиты данных в Германии.

RU.HETZNER.COM



100%
Green Electricity
Energy-efficient
Hardware

GreenIT **2011**
Best Practice Award

Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выбирайте более чистое будущее вместе с Hetzner Online!

* Нет платы за превышение. При превышении 20 ТБ/месяц (PX60), 30 ТБ/месяц (PX70-SSD) скорость соединения ограничивается (подсчёт ведётся по исходящему трафику, входящий и внутренний трафик не учитывается). Опционально можно снять ограничение, подтвердив оплату 80 рублей за каждый дополнительный ТБ.