

DVD ДАРОМ!



UBUNTU + FEDORA ОБНОВЛЕНЫ!

LINUX FORMAT

Главное в мире Linux

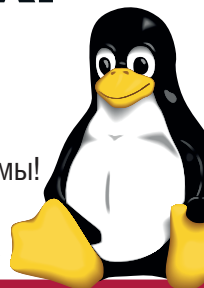
Апрель 2015 № 4 (195)

ПЕРЕХОДИТЕ НА LINUX!

Гид для новичка

- » Попробуйте сейчас
- » Наряду с Windows
- » Бесплатно программы!

плюс Рассмотрим три лучших дистрибутива



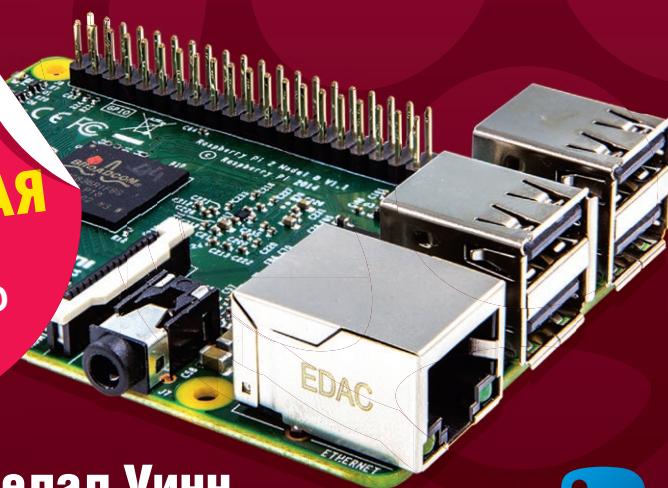
RASPBERRY PI 2

НА 500%
БЫСТРЕЕ

ЭКСКЛЮЗИВ! Практический гид и обзор новейшего Pi 2



ЕСТЬ
**ЭЛЕКТРОННАЯ
ВЕРСИЯ**
ДЛЯ ВАШЕГО
ПЛАНШЕТА!



Сделал Уинн

«Девизом сообщества Ruby стало «Путь к счастью — в оптимизации»»

Уинн Незерленд про работу с GitHub с. 34



Идем на Python 3!

» Начни программировать сейчас и портируй наработки



ghost

Блоггинг

Вникнем в Ghost

» Как установить платформу нового поколения



Вещание видео

Linux Motion

» Построим домашнюю систему видеонаблюдения

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 спросили наших экспертов: как они могут зажечь новое поколение линуксоидов энтузиазмом и желанием участвовать в свободном ПО?



Джонни Бидвелл

Увы, наш конкурент — Джастин Бибер Linux... Зато уже есть движение детей, изучающих программирование в школах. Совместная работа или драчка — вполне в духе СПО. Можно расти на примерах работ сверстников или разносить в клочья их идиотские методы. Прямо как в списках рассылки ядра.



Нейл Ботвик

Критическим здесь является Android. Людям, с малых лет принимающим как должное закрытую природу ПО и услуг, труднее объяснить пользу СПО. Покажите им, что они могут сами владеть своим оборудованием и изменять его, и они захотят этого навсегда.



Лес Паундер

Новое поколение пользователей СПО сейчас обучается на лучшем наглядном пособии: Raspberry Pi. Малобюджетный Linux-компьютер с СПО на борту столько всего умеет! Причем за смешную цену. Сочетание дешевого оборудования и кучи бесплатного ПО подсадит на себя и детей, и их родителей.



Маянк Шарма

Надо прокачать СПО через новейшие устройства. Итак, больше свободных приложений в магазине Steam, больше дистрибутивов Linux для Raspberry Pi и больше Android на переносных гаджетах. Да добрить все это разоблачениями а-ля Сноуден, чтобы люди не доверяли проприетарному ПО.



Александр Толстой

По-моему, следует подумать о среде разработки для детей — это здорово поможет. Как насчет кода на C++ для перетаскивания цветных кирпичиков и шариков без ввода с клавиатуры? Уверен: это создало бы у малышни должную дозу восторга насчет Linux, даже больше, чем замечательный TuxPaint!



Валентин Синицын

А? Что? Уже объявилось новое поколение?.. Вот, помню, в наши времена никого зажигать не надо было — ночами мы до красных глаз перекомпилировали ядро, настраивали X'ы... собирали KDE2 под FreeBSD...



С SQL и без него

» Уже четверть века для большинства ИТ-шников термин СУБД ассоциируется исключительно с реляционной структурой и языком SQL для работы с ней. Появление технологий NoSQL, о которых вы сможете прочесть на страницах этого номера, многими было воспринято как нечто революционное. Готов поспорить. По-моему, следует говорить о следующем шаге развития «хорошо забытого старого» — т.н. объектно-ориентированных СУБД, которые долгие годы оставались «в тени» реляционной парадигмы.

Можно привести пример области применения, где такой подход давно «победил»: службы каталогов. По сути, каталог является специализированной СУБД, хранящей информацию об объектах любого рода (например, пользователях и группах), каждый из которых имеет определенный набор атрибутов. Модель данных описана в стандарте ITU-T X.500, а общепринятым «языком запросов» служит LDAP. Поскольку многие атрибуты могут иметь не одно значение (пример: ACL, номера телефонов, членство в группах), «уложить» эти данные в набор таблиц весьма непросто. К тому же сама природа службы каталога требует, чтобы СУБД была распределенной и обеспечивала фоновую репликацию данных, соблюдая при этом требования ACID. Такие «движки» СУБД были созданы еще в начале 1990-х.

Рост популярности решений NoSQL лишний раз доказывает, что в реляционную схему отлично укладываются только бухгалтерские отчеты. Реальный мир «объектно-ориентирован», и без SQL во многих случаях просто удобнее.

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

Знаете всё о Linux?

ДА

НЕТ

Станьте автором
в журнале Linux Format!
linuxformat.ru/avtoram.phtml

Зарегистрируйтесь
на сайте
shop.linuxformat.ru



СОМНЕНИЯ

Скачайте бесплатно
архивные PDF-номера
журнала с сайта
linuxformat.ru/archive
и загляните на
wiki.linuxformat.ru



Версия для iPad
и iPhone доступна
в App Store

Выберите вид подписки

PDF-версия
на 6 месяцев

990 ₺

PDF-версия
на 12 месяцев

1800 ₺

Печатная версия
на 6 месяцев

1890 ₺

Печатная версия
на 12 месяцев

3480 ₺

Выберите вид доставки

Оплатите

Читайте Linux Format!

Станьте Linux-гуру

PDF-версия журнала Linux Format подойдет для тех, кто:

- Заботится о соблюдении прав деревьев
- Любит читать с экрана
- Мечтает получать каждый номер в день выхода журнала
- Хочет бесплатно скачивать содержимое DVD-приложения к каждому номеру

Печатная версия Linux Format понравится читателям, которые:

- Любят читать бумажные журналы
- Хотят получить в подарок подписку на PDF-версию Linux Format...
- ...а также диск с архивом журнала 2005–2014
- Порадуются новинкам открытого ПО на DVD-приложении к Linux Format в каждом номере

Способы доставки

- Курьером «ГНУ/Линуксцентра» по Москве и Петербургу
- Курьерской службой СПСР по России
- Почтой по России заказной или простой бандеролью
- Самовывоз из офиса «ГНУ/Линуксцентра» в Санкт-Петербурге
- Через пункты выдачи интернет-магазинов **iml.ru** в 11 городах России: Санкт-Петербург, Москва, Екатеринбург, Калуга, Нижний Новгород, Орел, Ростов-на-Дону, Тверь, Тюмень, Челябинск, Ярославль

Способы оплаты

- По квитанции в любом отделении Сбербанка
- Яндекс.Деньги, Webmoney
- Пластиковой картой Visa/MasterCard
- Наличными в офисе «ГНУ/Линуксцентра»
- Безналичный (для юридических лиц)



«ГНУ/Линуксцентр»
Санкт-Петербург,
пр. Медиков, 5, корп. 7
(метро «Петроградская»)
(812) 309-0686
www.linuxformat.ru

Содержание

«Больше тех, кто ценит пищу выше золота — миру веселее.» Дж. Толкиен

Обзоры

Firefox 35 12

Тестируем переотполированную версию браузера и его функцию глобальной коммуникации.

TigerVNC 1.4.0 13

Тигр, о тигр, светло горячий... достаточно ли функций в VNC, или им недали освещения?



➤ Не зовите это возвращением: *TigerVNC* выдал новый релиз.

Raspberry Pi 2 14

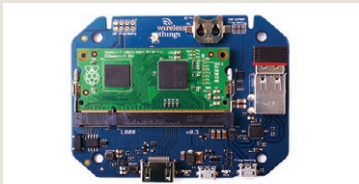
Из кремниевой печи восстал новый Raspberry Pi. Можно, наверное, утверждать, что мышь родила гору!



➤ Говорите, Raspberry Pi 2 вшестеро быстрее? Вот и проверим.

OpenPi 16

Новая примочка — с Raspberry Pi Compute в сердце и с открытой схематехникой. Wi-Fi!



➤ OpenPi подойдет предпринимателям с техническими идеями и уклоном.

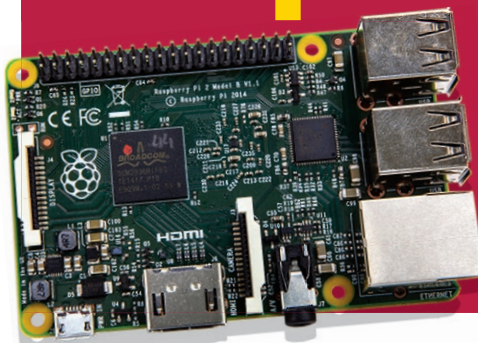
ThinkPad Yoga 11e 17

Со стоицизмом йога этот хромбук от Lenovo выдерживает ужасы школьной обстановки.

Civ: Beyond Earth 18

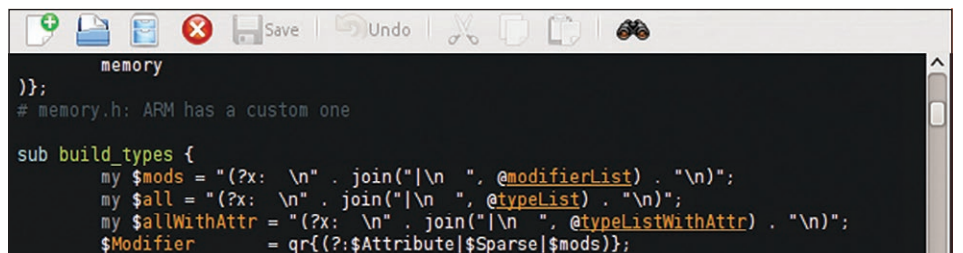
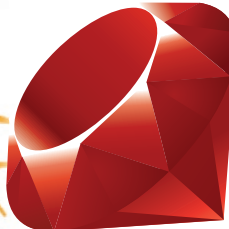
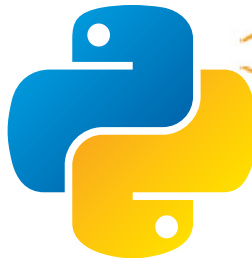
Просто очередная серия *Civilization 5* в космосе? Или есть в этой игре что-то особенное?

Новый Raspberry Pi 2 прибыл!



Подробный обзор с. 14
Практическое руководство с. 28

Сравнение:
Языки скриптов с. 22



Люди говорят



Мы жертвовали
открытостью Сети,
увеличивая число
«запертых садов»

Уинн Незерленд — о переменных в Web с. 34

На вашем бесплатном DVD



Fedora 21, Ubuntu 14.10, PCLinuxOS, ArchBang 2015 и другие!

» Убойные дистрибутивы по теме номера

ПЛЮС: HotPicks и бесплатная е-книга **с. 106**

Побалуйте себя и любимых подпиской на LXF!



Доступно в AppStore!



www.linuxformat.ru/subscribe

Ищите в этом номере

Переходим на Linux 40

Станем линуксоидами. И опробуем 3 лучших дистрибутива.

Riak — NoSQL 48

Как, совсем без SQL?! Да, админы это любят.

Безымянная Fedora 52

Хватит с нас Котов и Коров: функции главнее.

Apt бывают разные 56

Linux Mint завел свою, особенную утилиту *apt*.

Браузер Vivaldi 60

Не все радуются перемене курса в *Opera*.



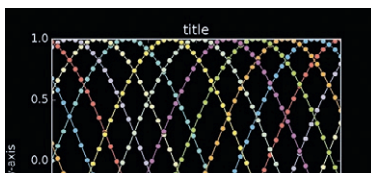
Академия кодига

Python 3 88

Быть может, ни одно продолжение не встречалось с меньшим энтузиазмом. Вы уж примиритесь со скобками...

Julia 92

Этот функциональный язык замечательно справляется с научными вычислениями, но и для общего назначения вполне пригоден.



Учебники

Minecraft/Pi Рисует черепашка 70

С черепашей помощью вы создадите в *Minecraft* на Raspberry Pi чудесную графику. Кто бы подумал, что такие деревья бывают...



» Разве же не артистичны черепахи?

ImageMagick Преобразуем картинки 74

Пустите свою мышь погулять. Манипулировать изображениями можно и без нее.

Motion Кто ел из моей миски? 76

Оборудуйте дома видеонаблюдение с web-камерой и Pi — муха незамеченной не пролетит.

Ghost Тема-сенсация 80

Дружелюбный учебник по созданию индивидуального оформления вашего блога.

1С для любимой ОС Пришествие 84

Знаменитая программа для бухгалтерии наконец-то явила благосклонность к Linux.

Постоянные рубрики

Новости 6 **Интервью LXF** 34 **HotPicks** 100

POCA нашла дистрибьютора и помогла химикам, HP открыла Distributed R, Стартап-тур ушел в загранку, Apple и MS наперебой обольщают пользователей, герои ядра отыскались, Лаборатория Касперского открыла вирус века, а злыдни перешли на личности.

Согласно Уинну Незерленду, оболочки API должны быть идиоматичны языку, на котором написаны. Ну вы поняли, о чем это он.

Отдайте горяченького! Лучшие в мире новинки свободного ПО: *Avalé, Boomaga, Dropbox Uploader, FFmpeg, Gloobus-preview, Hollywood, KWave, Linuxbrew, MPS-Youtube, SuperTuxKart, Yarock*.

Новости Android 20

Обратный ход: ОС с телефона идет на ПК, Windows просочилась в часы, G Watch гонятся за элегантностью, а уведомления доступны всем.

Рубрика сисадмина 64
Доктор созерцает восход и заход Солнца-Sun, вспоминает мышиную возню и сражается с коварными троянами.

Диск Linux Format 106

Содержимое двустороннего DVD этого месяца.

Сравнение 22

Есть ли жизнь после *Bash*? Мы сигналы 5 языков скриптов, готовых помочь измотанному сисадмину: *Bash, Perl 5, Python, Ruby, newLISP*.

Ответы 96

ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!
Нейл Ботвик — про менеджер пакетов в Fedora 21, горячие клавиши с дурным поведением, резервное копирование DVD, копирование LXF DVD на флэшки, загрузка по UEFI, исчезающие принтеры.

Пропустили номер? ... 108

Не предавайтесь бессильной злобе: его пока что можно купить.

Через месяц 112

Занявшись темой анонимности, мы достигнем таких высот, что своих не узнаем!



В ЭТОМ НОМЕРЕ: РОСА шагает по России » Большим данным легчает » Химики — за нас » Стартап-тур » Кошки-мышки гигантов » Лидеры ядра » Вирус-чемпион » Мишени атак

РОСА ШИРИТСЯ

На всей территории России

«Аксост» стал главным дистрибьютором для «НТЦ ИТ РОСА».



» Рубрику готовил
**АНДРЕЙ
ГОНДАРЕНКОВ**

По договору с компанией «НТЦ ИТ РОСА» компания «Аксост» получила эксклюзивные права на продажу программных продуктов «НТЦ ИТ РОСА» на территории РФ. Партнерам «Аксост» будет доступна для заказа вся линейка ПО: серверные и настольные ОС, включая сертифицированные ФСТЭК РФ и Минобороны РФ.

ОС от «НТЦ ИТ РОСА» все чаще выбираются крупными государственными и корпоративными заказчиками для построения защищенных автоматизированных систем. ОС РОСА «Кобальт», «Хром» и «Никель» в настольной и серверной редакции и разным уровнем сертификации предназначены для организаций, работающих с конфиденциальной

информацией, персональными данными и гостайной. Преимущества ОС от РОСА — удобный эргономичный интерфейс на основе лучших практик мирового уровня, быстродействие, совместимость с широким спектром прикладного ПО, разнообразие готовых ИТ-решений для различных задач совместно с партнерскими продуктами, регулярные обновления.

Гендиректор «НТЦ ИТ РОСА» Аркадий Тагиев отметил: «Растущая популярность наших защищенных продуктов потребовала выбора серьезного дистрибьютора, уже зарекомендовавшего себя на рынке и способного продвигать нашу продукцию в большом масштабе. „Аксост“ — перспективный и надежный партнер. Мы уверены, что подписание соглашения

с „Аксост“ позволит расширить использование продуктов „НТЦ ИТ РОСА“ для столь актуальных процессов импортозамещения». «ОС, предлагаемые „НТЦ ИТ РОСА“, представляют интерес для организаций самых разных отраслей и размеров, — говорит Михаил Прибочий, гендиректор „Аксост“. — Включение в портфель „Аксост“ решений российского вендора открывает новые возможности развития бизнеса для наших партнеров».

Компании начали сотрудничество с проведения ряда обучающих семинаров. В ближайших планах — активное взаимодействие в различных регионах РФ, направленное на популяризацию сертифицированных ОС РОСА и расширение дилерского канала.

ОСМЫСЛЯЕМ БОЛЬШИЕ ДАННЫЕ

Аналитика от HP

Язык Distributed R поддерживает и Vertica, и другие СУБД и платформы.

НР представила Haven Predictive Analytics, платформу с открытым кодом для крупномасштабного интеллектуального анализа в реальном времени, позволяющую работать с объемами данных на уровне петабайт. Её основа — новая версия языка статистического моделирования Distributed R. «Язык впервые предлагается в виде пакета с открытым кодом», пояснил вице-президент HP по маркетингу Джефф Вейс [Jeff Veis].

Язык программирования R применяется очень широко: это аппарат статистических вычислений и предикативной аналитики (как и ряд библиотек Python, SAS, MatLab,

Mathematica). Но классический R работает в один поток, что ограничивает объём данных для анализа. Учёным часто приходилось работать с выборками, а не со всем массивом данных, снижая точность выводов. Новые алгоритмы от HP Labs обеспечат запуск R-приложений сразу на нескольких компьютерах и обработку миллиардов строк, т.е. не выборки, а данных целиком.

Distributed R создан HP в первую очередь для аналитической СУБД Vertica, которая работает с терабайтами данных, и выпущен под открытой лицензией GPL v.2. Он ладит и с другими СУБД и платформами для обработки данных, типа Hadoop.

» Основной нового открытого решения HP является язык Distributed R.

Язык полностью совместим с R Studio и инструментами консоли разработчика R.

Здесь уместно вспомнить о покупке Microsoft компании Revolution Analytics, провайдера ПО и сервисов на базе языка R.



ИМПОРТОЗАМЕЩЕНИЕ

ROSA работает на химиков

Российский разработчик успешно заместил проприетарные приложения.

На фоне набирающего популярность импортозамещения многие предприятия сталкиваются с проблемами, существует ли отечественный аналог проприетарного продукта, насколько он способен заменить проверенное дорогостоящее решение, и т.д. Российский разработчик «НТЦ ИТ РОСА» готов поделиться опытом: дистрибутив ROSA Desktop Fresh успешно используется для обработки экспериментальных результатов, получаемых на оборудовании исследовательского Центра коллективного пользования (ЦКП) им. Д.И. Менделеева.

Для решения своих специфических задач химики используют ряд базовых программ:

- » представления периодической таблицы Менделеева — *gchemtable* и *gperiodic*;
- » «химический калькулятор» *Gchemcalc* (из набора *Gnome Chemistry Utils*) для анализа веществ сложной структуры;

» программы для рисования химических структур, трехмерных моделей молекул и кристаллов и т.п. Среди них немало свободных приложений — *rasmol*, *gcrystal*, *gchem3d* и др.

Для моделирования и анализа сложных структур специалистами из «РОСА» был установлен *Marvin Beans* от ChemAxon — закрытый инструментарий, имеющий вполне функциональную бесплатную версию. Написан инструментарий на Java, и в Linux чувствует себя отлично.

При обработке данных *LibreOffice Calc* оказался достойной заменой *MS Excel*. Для более сложных расчетов используются «продвинутое» приложения *Veusz*, *OpenDX*, система статистических вычислений R. В рассматриваемом случае оказалось, что основной кандидат на замену проприетарным аналогам — это *QtiPlot*, построенный по образу и подобию *Origin*, ибо последний являлся в организации

стандартом де-факто. Недостатком всех этих программ является то, что они не учитывают специфики обрабатываемых данных. Подходящей свободной программой такого рода является *massXpert*: это приложение предназначено для анализа спектров полимеров и при этом обладает требуемой степенью универсальности, поскольку не ограничивается каким-либо одним их видом.

«Главный итог наших усилий — это возможность использовать машину с Linux как рабочее место химика-аналитика. Отметим, что обойтись исключительно свободными программами для решения задач в других областях — задача для нас интересная. Надеемся, что дальнейшие российские разработки программных решений на основе свободного ПО дадут возможность ускорить процесс импортозамещения», прокомментировали в «НТЦ ИТ РОСА».

ДОРОГУ МОЛОДЫМ

Стартап-тур переходит границы

Из Всероссийского он становится Международным.

Стартап-тур впервые вышел за границы России: его очередной этап пройдет 19 и 20 марта 2015 года в Алматы (Казахстан), а 14 и 15 апреля его участников примет Минск (Белоруссия).

Целью Всероссийского стартап-тура, который проводится в 2015 году уже в четвертый раз, является поиск инновационных решений, востребованных на высокотехнологических рынках, и помощь молодым разработчикам в планировании собственной стратегии развития.

Одной из немаловажных задач Стартап-тура является также формирование инновационных сообществ, в которые вошли бы все, кто так или иначе связан с инновациями — сотрудники научно-исследовательских учреждений, студенты и преподаватели вузов, предприниматели, представляющие малый, средний и крупный высокотехнологический бизнес, венчурные инвесторы, представители органов власти.

Алматы станет шестым городом на маршруте Стартап-тура-2015: ранее его участников принимали Ростов-на-Дону, Томск, Нижний Новгород, Казань и Екатеринбург.

Центральной частью двухдневной программы Стартап-тура является конкурсный отбор проектов по следующим направлениям:

«Важная задача тура — формирование инновационных сообществ.»

- » информационные технологии;
- » биологические и медицинские технологии;
- » энергоэффективные и энергосберегающие технологии;
- » промышленные технологии и материалы;
- » инновации для детей.

В отборе могут принять участие все желающие — как юридические, так и физические лица: малые инновационные компании, научные коллективы, индивидуальные ученые, предприниматели, студенты. К рассмотрению принимаются проекты на любой стадии развития — от идеи до готового продукта. Участникам Стартап-тура предоставляется возможность выступить с презентацией своих проектов перед экспертами и потенциальными инвесторами.

По каждому из указанных направлений выявляются три победителя, которые, помимо призов от партнеров и со-организаторов Стартап-тура, получают приглашение на крупнейшую российскую конференцию стартапов — Startup Village (Москва, 2 и 3 июня 2015 года, Инновационный центр «Сколково»).

В программу Стартап-тура входят также мастер-классы от ведущих специалистов в области реализации инновационных проектов, семинары и тренинги.

ВАЖНЫЕ ОТКРЫТИЯ

Конкуренция на благо народу

Для пользователей Linux открыты Apple iWork и Microsoft Hadoop on Azure.

Конкуренция между двумя крупнейшими мировыми технологическими гигантами очень продуктивна для пользователей. II квартал февраля ознаменовался двумя интересными событиями. Сначала Apple выдала в доступ пользователей других платформ, в том числе Windows, Linux, Chrome OS и Android, бета-версию своего сервиса iCloud. После регистрации на iCloud из под «чужеродной» ОС её пользователь получает 1 ГБ хранилища и доступ к Mail, Contacts, Calendar, Drive, Reminders, Pages, Numbers и Keynote. Затем на конференции Strata+Hadoop World в Сан-Хосе Microsoft анонсировала предварительную Linux-версию сервиса HDInsight (Hadoop on Azure). Подобно аналогу для Windows, HDInsight для Linux создан поверх Hortonworks Data Platform (HDP) и обеспечивает полную совместимость с Apache Hadoop плюс интеграцию с инструментами

бизнес-аналитики Microsoft (*Excel, SQL Server, PowerBI*). «Мы хотим сделать Azure лучшим местом для хранения данных», заявил вице-президент Microsoft Т.К. Ренгараджан [Т.К. «Ranga» Rengarajan].

Что это? Конкуренция между крупнейшими мировыми технологическими гигантами возвращается после нескольких лет дремоты, сотрудничества и сладких слов?

ПО и онлайн-сервисы Microsoft для других платформ широко распространены и доступны практически всем (за подписку на web-приложения пакета Office 365 взимается достаточно скромная месячная плата, а Office Online вообще дармовой). И вот Apple пытается поколебать позиции Microsoft. Облачный сервис Apple сейчас бесплатен, но нет никаких гарантий, что в будущем положение не изменится. Или же Apple поняла, что нуждается в сообществе «не-Apple» в большей степени, чем в сохранении эксклюзивности?

Вряд ли iTunes и iMessage станут в ближайшее время доступны широкой аудитории; а на рабочем месте серьёзной фирмы iWork так же бесполезен, как и iPhone; но, в любом случае, это важный шаг в направлении кросс-совместимости в проприетарном мире. На фоне этих событий недавнее обновление *Microsoft Office* для iPhone и iPad бесплатно предоставило дополнительную функциональность.

В такой игре в «кошки-мышки» пользователи безусловно заинтересованы.

➤ После регистрации на iCloud пользователь «чужеродной» ОС получает 1 ГБ хранилища.



ЯДРО LINUX

Они сражались за Linux

Отчёт The Linux Foundation показал, кто создаёт ядро нашей любимой ОС.

Без сомнения, среди приводящих в действие современную ИТ-инфраструктуру технологий, наиболее используемой в мире является Linux. От дронов до космических станций, от гигантских суперкомпьютерных кластеров до крошечных умных часов, от оборудования фондовых бирж до домашних принтеров и маршрутизаторов, все приводится в действие ОС GNU/Linux.

Linux — повсюду, и Linux — вездесущ. Кроме того, Linux — крупнейшая из совместно используемых, совместно разрабатываемых и имеющих открытый исходный код технологий. Кто же выполняет всю работу? Кто пишет код, которым все мы, даже приверженцы Apple или Windows, пользуемся, зачастую и не подозревая об этом? Вот какие факты содержатся в ежегодном отчёте The Linux Foundation. Статистика, ведущаяся с 2005 г., зафиксировала за это время вклад в работу над ядром от около 11 800 программистов из более 1200 компаний. За 2014 г.

добавилось свыше 1400 новых разработчиков и 200 компаний, было внесено 96 тыс. изменений от 4171 разработчиков из 518 компаний. Наибольшее число изменений (13722) за истекший год внесено в ядро версии 3.15.

По данным последнего отчёта, лидером среди корпоративных контрибью-

«Лидером среди корпоративных контрибьюторов оказалась Intel.»

торов, как по общему вкладу в развитие ядра, так и по количеству привлеченных новых разработчиков, оказалась компания Intel (147 новых участников). Другие лидеры по вкладу — Red Hat, Linaro, Samsung, IBM, SUSE, TI, Vision, Google и Rareness. Из прочих известных компаний в списке присутствуют Oracle (14-е место), AMD (15), NVIDIA (16), ARM (19), Cisco (21).

Среди причин, по которым компании принимают участие в разработке ядра Linux, названы:

- » использование Linux на своём оборудовании (IBM, AMD, ARM, Intel, SGI, MIPS, Freescale, HP, Broadcom, NVIDIA, Fujitsu);
- » заинтересованность в повышении функциональности ядра (Red Hat, Novell, Oracle, MontaVista);
- » использование Linux в качестве основы программного обеспечения своих продуктов (Sony, Cisco, Samsung).

А вот вклад неоплачиваемых разработчиков в последние годы неуклонно сокращался: 14,6% в 2012 году, 13,6% в 2013-м, 11,8% в 2014-м.

Возросла доля участвующих в создании ядра женщин: за минувший год программа FOSS Outreach Program for Women предоставила сообществу разработчиков 24 представительницы прекрасного пола. Спонсорами программы стипендий за разработку ядра были Codethink (1), Intel (3) и Linux Foundation (3).

ГЛОБАЛЬНЫЙ КИБЕР-ШПИОН

Вирус всех основных ОС

«Лаборатория Касперского» сочла его сильнейшим в мире.

Эксперты «Лаборатории Касперского» изучили более 60 операций кибер-шпионажа по всему миру; однако недавно выявленная превзошла по масштабам, инструментам и эффективности все ранее известные. Кибер-группа, названная Equation Group, действует почти 20 лет и кооперировалась с организациями нашедших атак Stuxnet и Flame, затронув десятки тысяч пользователей в более чем 30 странах. Наибольшее число жертв зафиксировано в России и Иране. Инфраструктура Equation Group включает более 300 доменов и 100 серверов управления, в т.ч. в США, Великобритании, Италии, Германии, Нидерландах, Панаме, Коста-Рике, Малайзии, Колумбии и Чехии. «Лаборатория Касперского» ныне контролирует около 20 серверов группы.

Уловки в арсенале Equation Group множество, и некоторые из них буквально новаторские. «Лаборатория Касперского»

впервые за свою практику обнаружила модули, изменяющие заводскую прошивку жестких дисков 12 основных производителей, добиваясь этим двух целей: во-первых, попав на жесткий диск, зловред остается там навсегда — его нельзя ни обнаружить, ни избавиться от него, даже если отформатировать диск. Во-вторых, создается «тихая гавань» — секретное хранилище, где безопасно копится необходимая информация.

Equation Group также применяет червя Fanny, способного ограбить компьютер, даже не связанный с Интернетом. С уже зараженной машины червь переходит на USB-флэшку, где создает скрытый сектор и собирает в нём информацию об архитектуре сети. Нашував компьютер с Интернет-соединением, червь с USB передает данные серверу Equation Group. Атакующие могут добавлять в тот же скрытый сектор команды, и червь их выполнит.

ИНТЕРЕС К ЛЮДЯМ

IT-безопасность-2014

Кто-то теряет, а кто-то находит...

В 2014 г. худшие для безопасности IT-систем варианты развития событий стали банальным явлением. Сообщения об утечке данных, вероятно, не прекратятся, поскольку бизнес, правительства и регулирующие органы не успевают адекватно реагировать на угрозы.

Компания Gemalto отмечает, что трендом года стала кража персональных данных, в т.ч. доверенных солидным учреждениям. «Breach Level Index», годовой отчет компании, гласит, что итоги выявленных в 2014 г. 1514 инцидентов — компрометация данных свыше 1 млрд пользователей: рост против 2013 г. на 78%. Это 54% всех атак (на 20% больше, чем в 2013 г.). Примеры — хищение записей компании по медицинскому страхованию Anthem (80 млн), Home Depot (109 млн), eBay (145 млн) и JP Morgan Chase (83 млн). Сотрудник Gemalto Цион Гонен [Tsion Gonen] объясняет это ростом оперативности компаний по финансовым услугам:

взломанная банковская карта сразу блокируется, а данные об идентификации пользователя дают кибер-ворам гораздо более широкие возможности. Инцидентам с Home Depot, eBay и JP Morgan Chase присвоена оценка 10 (высшая по шкале Gemalto), а, скажем, инциденту с кражей у компании Sony цифрового контента — только 6,5, так как наибольший ущерб нанесен самой компании.

55% утечек данных в 2014 г. — дело рук хакеров; 25% вызваны человеческой ошибкой (неправильной организацией защиты сайтов с доступом к данным о клиентах, потерей ноутбука и т.п.). Решение компаний не шифровать данные клиентов человеческой ошибкой не считается.

Тем не менее, по мнению Гонена, 2014 г. стал переломным для IT-безопасности: резко выросла осведомленность о проблемах; в ряде стран, включая США, приняты новые законы о кибербезопасности; на новый уровень выходит сотрудничество государства с частными фирмами. **LXF**

Новости короткой строкой

С 14 июля 2015 Microsoft прекратит поддержку ОС Windows Server 2003. Для корпоративных пользователей это станет ещё одним аргументом для перевода своих серверов на GNU/Linux. Источник: www.microsoft.com

Релиз *systemd 219* содержит ряд улучшений в *networkd*; поддержке изолированных контейнеров; снимках и подразделах *Btrfs*. Источник: lists.freedesktop.org

Линус Торвалдс изменил нумерацию ядра (выпустив первый релиз-кандидат Linux 4.0), а также кодовое имя ядра — с «Diseased Newt» на «Hurr durrr I'ma sheep». Источник: lkml.org

Благодаря кампании на Kickstarter по совместному финансированию разработки, новый выпуск растрового графического редактора *Krita 2.9* стал самым ярким релизом в истории проекта. Источник: krita.org

Xfce 4.12, новый стабильный выпуск свободной среды рабочего стола для UNIX-подобных ОС, представлен через 2 года и 10 месяцев после предшественника. Этот срок намного превысил планы разработчиков. Источник: www.xfce.org

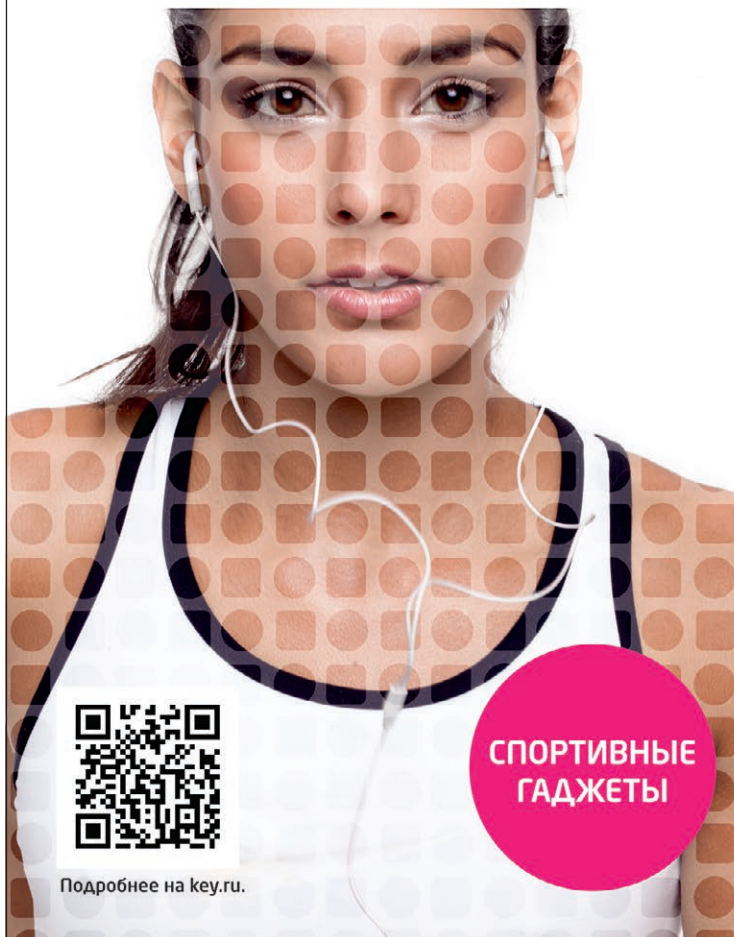
После полутора лет разработки вышел очередной релиз многофункционального серверного продукта *Samba 4.2.0*, с одновременным прекращением поддержки ветки *Samba 3*. Источник: lists.samba.org

Microsoft открыла код используемой для сборки приложений в Visual Studio и .NET-утилиты *MSBuild* (Microsoft Build Engine), и планирует портировать её на GNU/Linux и Mac OS X. Источник: www.microsoft.com

Новый проект *Nuntius*, ориентированный на Gnome 3.16 с его переработанной системой уведомлений, обеспечивает через Bluetooth вывод уведомлений с Android-устройств (WhatsApp или SMS) на рабочий стол ПК. Источник: blogs.gnome.org

ФАС России по жалобе «Яндекса» возбудила дело в отношении компании Google Inc. по признакам нарушения антимонопольного законодательства РФ (ст. 14 ФЗ «О защите конкуренции» и, возможно, некоторых других статей 135-ФЗ). Источник: www.fas.gov.ru

ТВОИ ПОМОЩНИКИ В ТРЕНИРОВКАХ



Экшн - камера



Экшн-камера Sony HDR-AS30

Smart - часы

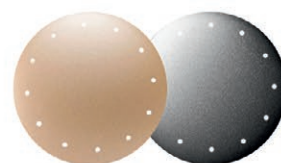


Smart-часы Cookoo Watch

Smart - здоровье



Браслет Jawbone UP



Шаромер Misfit Shine



Техника и человек

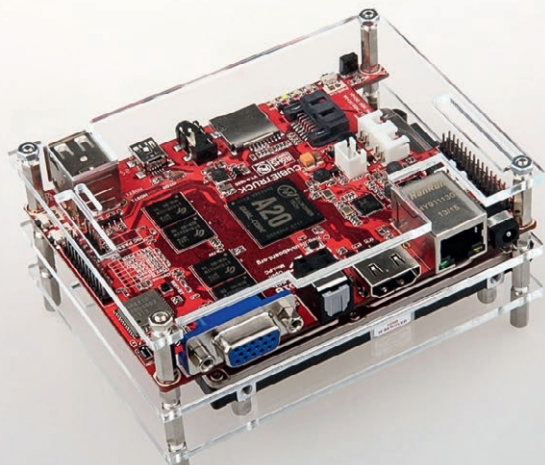
СПОРТИВНЫЕ
ГАДЖЕТЫ

Подробнее на key.ru.

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

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

КОМПЛЕКТАЦИЯ

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

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

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





Новинки программного и аппаратного обеспечения в описании наших экспертов



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

SverigeLinux как замена импорта

Тема импортозамещения нынче актуальна как никогда, и ни одно издание, вне зависимости от профиля, пройти мимо неё не может. Как и **LXF**. Тем более что давеча про просторам Интернета разнеслась весть о представлении первой альфа-версии дистрибутива SverigeLinux, на базе Debian. Он имеет шведское происхождение, разработка его финансируется Шведским интернет-фондом, и предназначен он для шведских госучреждений. Какое отношение он имеет к нашему импортозамещению?

Дистрибутив позиционируется как система, которая может быть быстро развёрнута администраторами, имеющими минимальные познания в Linux'e. Что даёт возможность ответить контр-инициативой: разработкой дистрибутива, устанавливаемого на тысячах рабочих мест администраторами, не имеющими никаких познаний в этой ОС. Желательно, чтобы его можно было установить, не устанавливая. И уж совсем в идеале — чтобы с ним и работать не требовалось. Хотя нечто подобное у нас уже было?..

А если серьёзно, то ничего необычного в SverigeLinux нет: подобные системы с давних пор разрабатываются в разных провинциях Испании (и некоторых других стран). Нет тут ничего сверхъестественного и технически — для этого имеется достаточно штатных инструментов. Вот только финансируются такие работы обычно заинтересованными организациями, а не ведутся на голом энтузиазме, подобно Matuntu, о котором говорилось в прошлом номере.
alv@posix.ru

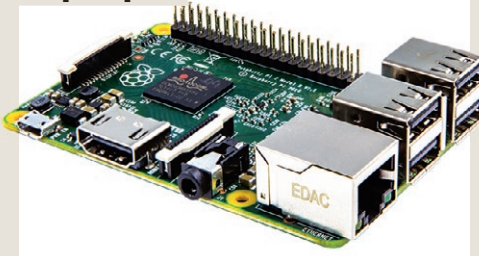
Сегодня мы рассматриваем:

- Firefox 35** 12
Отпраздновав десятилетие, Огнелис выпустил новую версию. Вы удивитесь: мало ему динамического изменения стилей, так ещё и его дополнение Hello затевает утереть нос Skype.
- OpenPi** 16
Наш ответ модной теме Интернета Вещей, и притом с открытыми спецификациями: беспроводной хаб, укрепляемый поверх вычислительного модуля Raspberry Pi.
- TigerVNC 1.4.0** 13
Некогда знаменитый VNC-сервер начал было загнивать, но ожил и разразился главным релизом. Работает на ожидаемом уровне, но лучше всего — со своим же клиентом.
- ThinkPad Yoga 11e** 17
Злые школьники способны запинать любой хромбук, но только не этот! Lenovo позаботилась о солидном запасе прочности. Кстати, поворотом экрана он превращается в планшет.
- Raspberry Pi 2** 14
Компьютер-крошка и раньше был вполне полноценным, а его второе издание вообще обзавелось 1 Гб ОЗУ. Заодно сбылись многие мечты сообщества насчет улучшений.
- Civ: Вне Земли** 18
Любите *Civilization*? Получайте расширение этой игры на просторы космоса. Это не просто сиквел, а во многом самобытное творение, способное вас удивить.



Хромбук от Lenovo отличается минималистской эстетикой.

Raspberry Pi 2



Процессор нового Raspberry Pi — четырехъядерный, что и обеспечивает прирост скорости.

Civ: Beyond Earth



Когда вам надоест бороздить просторы космоса, откройте портал Освобождения — и до дому.

Сравнение: Языки скриптов с. 22

Bash

```
#!/bin/bash
set -e

function delete_dir {
    local dir="$1"
    if [ -d "$dir" ]; then
        rm -rf "$dir"
    fi
}

delete_dir "$@"
```

newLISP

```
(apply-dir delete-file remove-dir "someDir")

; the example would delete all file and directories
; in someDir in the current directory

(define (apply-dir ffunc dfunc dir)
  (do-list (nde (directory dir))
    (if (A (directory?) (append dir "/" nde))
        (= nde ".") (= nde ".."))
    (begin
      (apply-dir (func dfunc (append dir "/" nde))
        (dfunc (append dir "/" nde)))))
```

Perl

```
#!/usr/bin/perl
use strict;
use warnings;

my $dir = shift;
my $delete = 1;

sub build_tree {
    my $dir = shift;
    my @files = glob("$dir/*");
    my @dirs = ();
    my @files_to_delete = ();
    my @dirs_to_delete = ();

    for my $file (@files) {
        if ($file =~ /\.*/i) {
            push @files_to_delete, $file;
        } else {
            push @dirs_to_delete, $file;
        }
    }

    for my $dir (@dirs_to_delete) {
        rmdir($dir);
    }

    for my $file (@files_to_delete) {
        unlink($file);
    }
}
```

Python

```
import time

def modified_within(top, seconds):
    now = time.time()
    for path, dirs, files in os.walk(top):
        for name in files:
            fullpath = os.path.join(path, name)
            if os.path.exists(fullpath):
                atime = os.path.getatime(fullpath)
                if atime > (now - seconds):
                    print(fullpath)

if __name__ == '__main__':
```

Ruby

```
#!/usr/bin/ruby
require 'open3'

def delete_dir(dir)
  Dir.glob("#{dir}/*").each do |file|
    if File.directory?(file)
      delete_dir(file)
    else
      File.delete(file)
    end
  end
end

def main
  delete_dir(ARGV[0])
end

main
```

Скрипты существенно упрощают жизнь пользователей и сисадминов, и желательно найти общий язык с системой, чтобы эти скрипты органично в нее вписались. Что ж, поищем...

Firefox 35

Маянк Шарма тихонько напевает новую версию песни *Hello* Лайонела Ричи [Lionel Richie], развлекаясь с голосовыми и видеовозможностями нового *Firefox*.

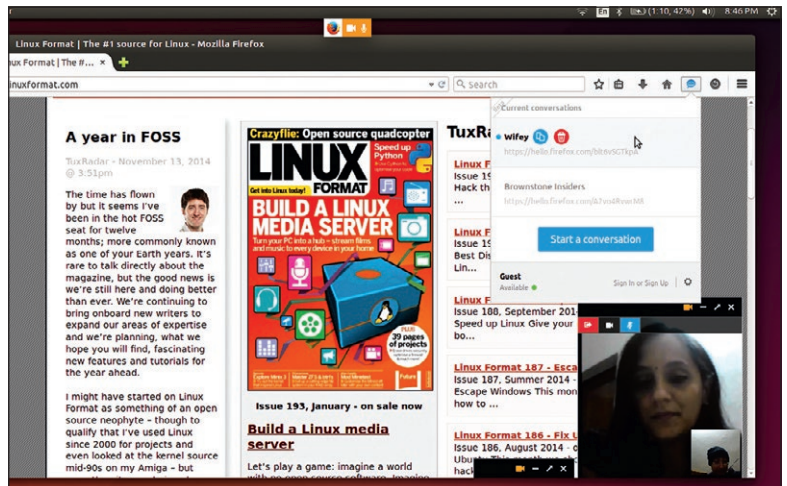
Вкратце

» Один из самых популярных и богатых функциями браузеров с открытым кодом. См. также: *Chromium*.

Linux отнюдь не испытывает недостатка в качественных web-браузерах. Но несмотря на насыщение, *Firefox* сумел остаться в лидерах гонки, благодаря непрерывному совершенствованию и инновационным функциям-дополнениям. *Firefox 35*, последний релиз от Mozilla [на момент написания статьи, — прим. пер.], не то что ломается от новых функций, но даёт пользователям шансы впитать все добавленные в предыдущих релизах.

Несомненно, самое заметное в этом выпуске — возможность делать полноценные голосовые и видеозвонки, получившая в *Firefox* название Hello. Mozilla называет её «первой глобальной системой общения, встроенной прямо в браузер». Хотя это всё ещё бета, она неплохо работает и выполняет всё обещанное. Mozilla в первую очередь стремилась дать пользователям общаться, не сдавая личную информацию третьим лицам. Мало того, Mozilla хотела демократизировать удобство бесплатных сервисов общения, поэтому вы можете созваниваться с любым WebRTC-совместимым браузером, таким как *Chrome*, *Opera* и сам *Firefox*, а также между любыми настольными и мобильными платформами.

Mozilla впервые представила экспериментальную поддержку WebRTC в *Firefox 33* и официально запустила её в *Firefox 34*, используя платформу связи в реальном времени OpenTok от TokBox. В нынешнем выпуске Mozilla улучшила и упростила процесс совершения и получения звонков.



» Управление приватностью в *Firefox* распространяется и на Hello. Вы можете отключать камеру или микрофон (одним щелчком) или приём любых звонков.

Чтобы позвонить, надо всего лишь щелкнуть по иконке Hello на панели *Firefox*, а затем нажать на кнопку Сделайте звонок. Сгенерируется ссылка на комнату общения [chat room]; ссылку можно передать любому, с кем вы хотите поговорить. При нажатии на ссылку для начала разговора появится окно с вашей комнатой. Когда приглашённый вами человек тоже щёлкает по ссылке, он добавляется в комнату, а иконка Hello загорается, чтобы уведомить вас.

Позвони мне, позвони

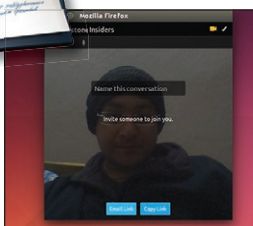
Hello позволяет также создать несколько комнат, каждую со своим уникальным URL. Задно вы можете дать диалогу имя и сохранить его для быстрого доступа к людям, с которыми общаетесь регулярно. А мы не забыли упомянуть, что всё это делается без настройки какой-либо учётной записи? Впрочем, если вы заведёте себе учётную запись *Firefox*, то сможете добавлять или импортировать существующие контакты электронной почты. Добавление довольно банально, а вот для импорта из Google потребуются указать браузеру OAuth-данные вашей учётной записи Google. Когда вы их добавите, контакты можно будет редактировать, блокировать или удалять, щелкая по стрелочке возле их названий.

Если вы считаете эту функцию скорее рекламной фишкой, чем реальной заменой программ для видеозвонков типа *Skype* или *Facetime*, то мы думаем, что

результаты вас удивят. Hello *Firefox* — серьёзный шаг к замене настольных программ web-решениями. Он богат возможностями, и в нашем тесте работал между разными браузерами на разных ОС, и даже между устройствами на Android.

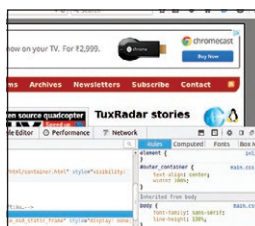
Помимо Hello, в этом выпуске есть и другие изменения. В заметках к выпуску сказано, что браузер теперь гораздо лучше подготовлен к динамическим изменениям стилей, и в нём реализовано новое расширение закрепления публичных ключей [public key pinning], которое должно улучшить аутентификацию защищённых соединений. Релиз, кроме того, содержит несколько улучшений для web-разработчиков, таких как включённые по умолчанию CSS-фильтры. **LXF**

Свойства новскидку



Firefox Hello улучшен

Процесс совершения звонка упростился и теперь использует модель диалогов, которая основана на комнатах.



Для web-разработчиков

Более 150 поправок в инструментах и новые функции CSS, например, свойства filter и сопоставление исходников [source maps].

LINUX FORMAT **Вердикт**

Firefox 35

Разработчик: Mozilla Foundation
 Сайт: www.firefox.com
 Лицензия: Mozilla Public Licence

Функциональность	9/10
Производительность	9/10
Удобство в работе	9/10
Документация	9/10

» Более стабильные видеозвонки и традиционное изобилие обновлений безопасности делают обновление *Firefox* полезным.

Рейтинг 9/10

TigerVNC 1.4.0

Маянк Шарма проверяет, заслужила ли новая версия знаменитого в прошлом VNC-сервера ещё один шанс.

Вкратце

» Клиент и сервер удалённого рабочего стола, использующие протокол VNC. См. также: *Vino* и *Krfb/Krdc*.

Половину десятилетия назад, в 2009-м, сервер *TigerVNC* при поддержке Red Hat ответил на *TightVNC*. Целью проекта было сделать возможным запуск ресурсоёмких 3D- и видеоприложений в полноэкранном режиме на удалённом компьютере. Вскоре после дебюта *TigerVNC* был принят в дистрибутив Fedora как реализация VNC по умолчанию. С годами проект сбавил ход, так как некоторые разработчики сбежали с корабля. Выпуски *TigerVNC* стали реже. Но недавно проект объявил о новом главном релизе, и нам захотелось посмотреть, как он смотрится на фоне конкурентов.

TigerVNC позаимствовал некоторые примечательные особенности у других проектов — например, аппаратное сжатие JPEG. Помимо Linux, проект запускается на Windows и Mac OS X. *TigerVNC* включает скучный кроссплатформенный просмотрщик, не содержащий функций, доступных в других клиентах, таких как *Remmina* и *Vinagre*. В нём нет ни списка запущенных серверов VNC в сети, ни возможности добавлять соединения в закладки. Взамен вы получаете простое окно с текстовым полем для ввода IP-адреса сервера, к которому хотите подключиться.

Но это не означает, что в нём нет настроек. Щёлкните по кнопке Параметры, и вы найдёте массу параметров для настройки соединения. По умолчанию программа настроена на автоматический выбор наилучших параметров для вашего соединения. Это очень мило, если учесть,



» Проект *VirtualGL* опубликовал подробный отчёт об этом релизе, обнаружив серьёзную потерю производительности.

что большинство таких параметров понятны только бывалому сисадмину, привыкшему к принципам удалённых рабочих столов. Вы можете указать такие вещи, как глубина цвета и степень сжатия, выбрать схему аутентификации и механизм шифрования и т.д.

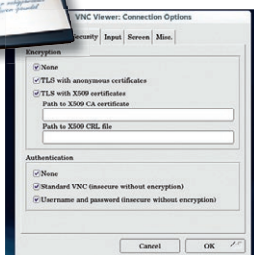
Существует также сервер *Xvnc* — он создаёт собственную виртуальную X-сессию, к которой могут подключаться удалённые пользователи. С другой стороны, если вы хотите подключиться к существующей X-сессии, есть также сервер *x0vncserver*. В последней версии он стал поддерживать расширение *XDamage* для отслеживания изменений, которое, согласно официальным заметкам к выпуску, делает сервер «чуть менее бесполезным».

подключиться к серверу *TigerVNC* с других клиентов, в паре с собственным клиентом он работает лучше всего, не выбрасывая неприятных ошибок. Удивительно, но *TigerVNC* был более отзывчив при просмотре удалённой сессии Windows, чем сессии другого дистрибутива Linux. Более того, показ удалённого рабочего стола Gnome потребовал больше времени и усилий, чем хотелось бы затрачивать среднестатистическому пользователю.

В целом, в этом выпуске не обнаружилось особо выдающихся функций. Собственно говоря, если хорошенько поразмыслить, *TigerVNC* не предлагает каких-либо уникальных возможностей, которые заставили бы нас порекомендовать его вместо других клиентов, скажем, *Vino* или *krfb*. **LXF**

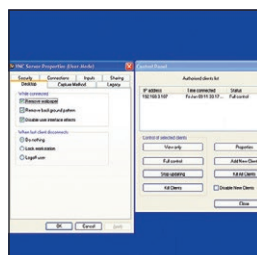


Свойства навскидку



Уборка кода по весне

Помимо исправления ошибок, был улучшен код и доработаны некоторые существующие функции.



Кросс-платформенный

Клиенты для Windows и Mac OS X также были улучшены для лучшего пользовательского опыта.

Беззубый тигр?

На самом деле последний выпуск полон подобных мелких правок и улучшений, в основном скрытых от глаз. Помимо множества исправлений, разработчики произвели закулисные улучшения работы с клавиатурой как на сервере, так и на клиенте, а также улучшили реализацию TLS в Java-клиенте проекта. Серверный компонент нового релиза стал лучше поддерживать последние версии сервера *X.org* и избавился от поддержки индексированного 256-цветного режима. Функция, которая не прошла дальше бета-версии — поддержка IPv6: её отложили в связи с необходимостью дополнительных доработок.

В наших тестах *TigerVNC* работал на ожидаемом уровне. Хотя вы можете

LINUX FORMAT Вердикт

TigerVNC 1.4.0

Разработчик: Команда TigerVNC
Сайт: www.tigervnc.org
Лицензия: GPL

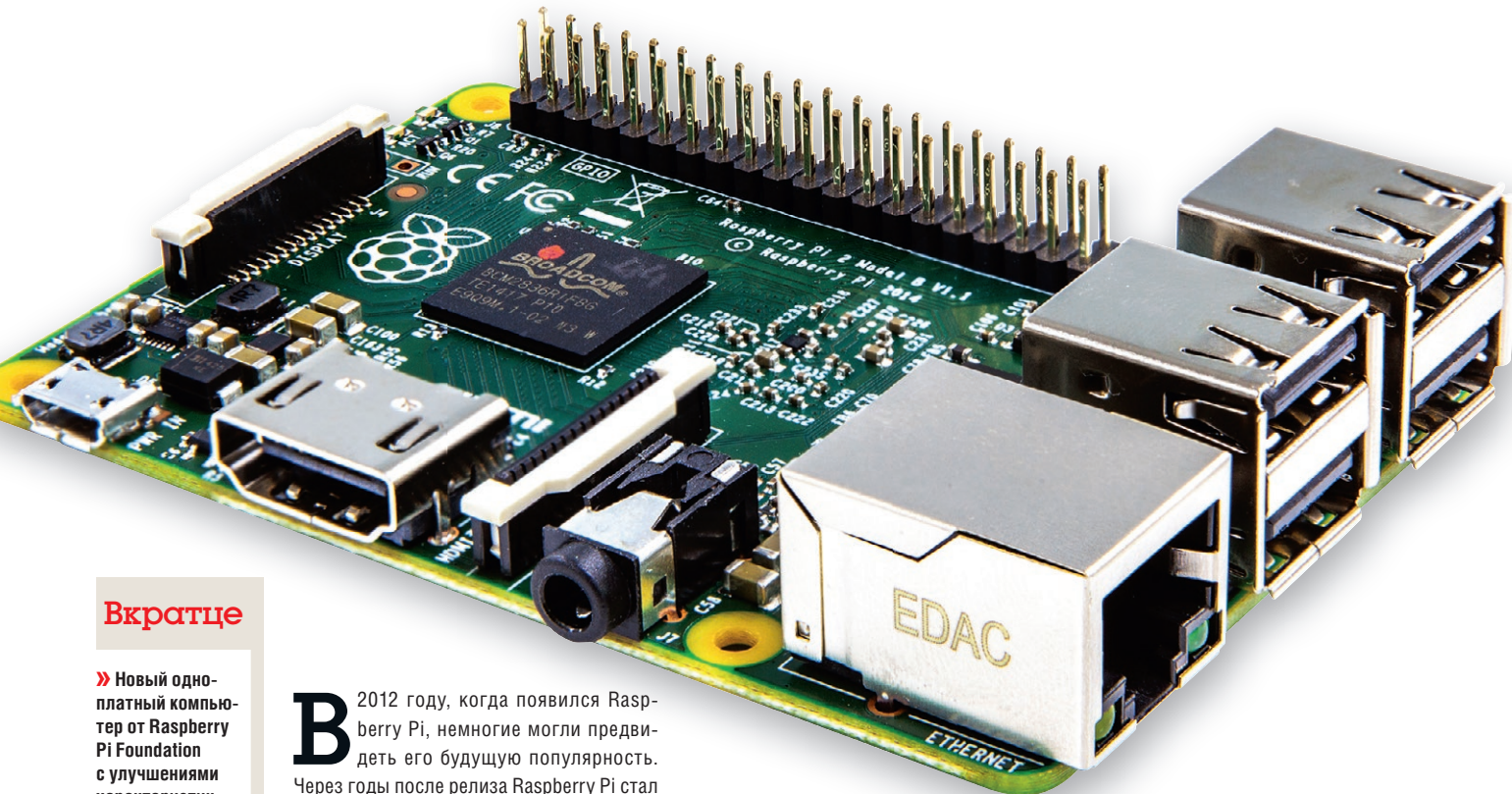
Функциональность	5/10
Производительность	5/10
Удобство в работе	4/10
Документация	5/10

» Средний релиз среднего приложения, которое работает, но без особых преимуществ по сравнению с соперниками.

Рейтинг 5/10

Raspberry Pi 2

Лес Паундер облизывается, глядя на новый Pi, и спешно вонзает зубы в новое сладкое угощение со вкусом малины.



Вкратце

» Новый одноплатный компьютер от Raspberry Pi Foundation с улучшениями характеристик, о которых мы мечтали. Pi 2 — самый последний в линейке продуктов от Pi Foundation, и на нём может работать ряд дистрибутивов Linux.

Спецификация

- » Чип: Broadcom 2835
- » Процессор: 4-ядерный ARM7 на 800 МГц
- » Графика: Videocore IV 250 МГц
- » Память: 1 Гб
- » GPIO: 40 контактов
- » Разъёмы: 4×USB 2.0, 100BaseT Ethernet, HDMI, MicroSD
- » Габариты: 85,60×56,5 мм

В 2012 году, когда появился Raspberry Pi, немногие могли предвидеть его будущую популярность. Через годы после релиза Raspberry Pi стал самым известным одноплатным компьютером на рынке и породил множество подражателей, но ни одного — с таким богатым сообществом, которое органично выросло вокруг Raspberry Pi.

Со времени релиза исходного Raspberry Pi успело выйти три версии флагманской модели B, начавшей с 256 Мб ОЗУ и дошедшей до 512 Мб у второго B и B+. Но во всех этих моделях системой на кристалле [system-on-a-chip, SoC] оставался надёжный BCM2835 с процессором ARM11 на 700 МГц. Сообщество делало потрясающие вещи даже и с этими ресурсами, а ныне пришли долгожданные улучшения характеристик.

В начале февраля вышел Raspberry Pi 2, где прежний ARM11 был заменён на ARM 7, работающий на увеличенной частоте 800 МГц. И вместо одного, новая версия получила четыре ядра, что увеличивает скорость Raspberry Pi ни много ни мало вшестеро. Чтобы не посрамить новый процессор, объём ОЗУ увеличили до 1 Гб. Остальное железо соответствует таковому от B+: 40-контактный GPIO, четыре порта USB2 и Ethernet 10/100. Физически Raspberry Pi 2 имеет такие же размеры, что и B+.

На тестовом стенде

Чтобы показать улучшения, сделанные в Pi 2, мы решили запустить несколько испытаний из реального мира и выявить, насколько новый Pi мощный в прямом сравнении с B+.

Первым тестом в нашем списке была холодная загрузка обоих Pi до приглашения входа. B+ справился с этим за 33 секунды против 17 секунд у Raspberry Pi 2.

«Ныне пришли долгожданные улучшения характеристик.»

Затем мы загрузились на обоих Pi до рабочего стола; у B+ это заняло 42 секунды, а у Pi 2 — 21 секунду, вдвое меньше, чем у B+!

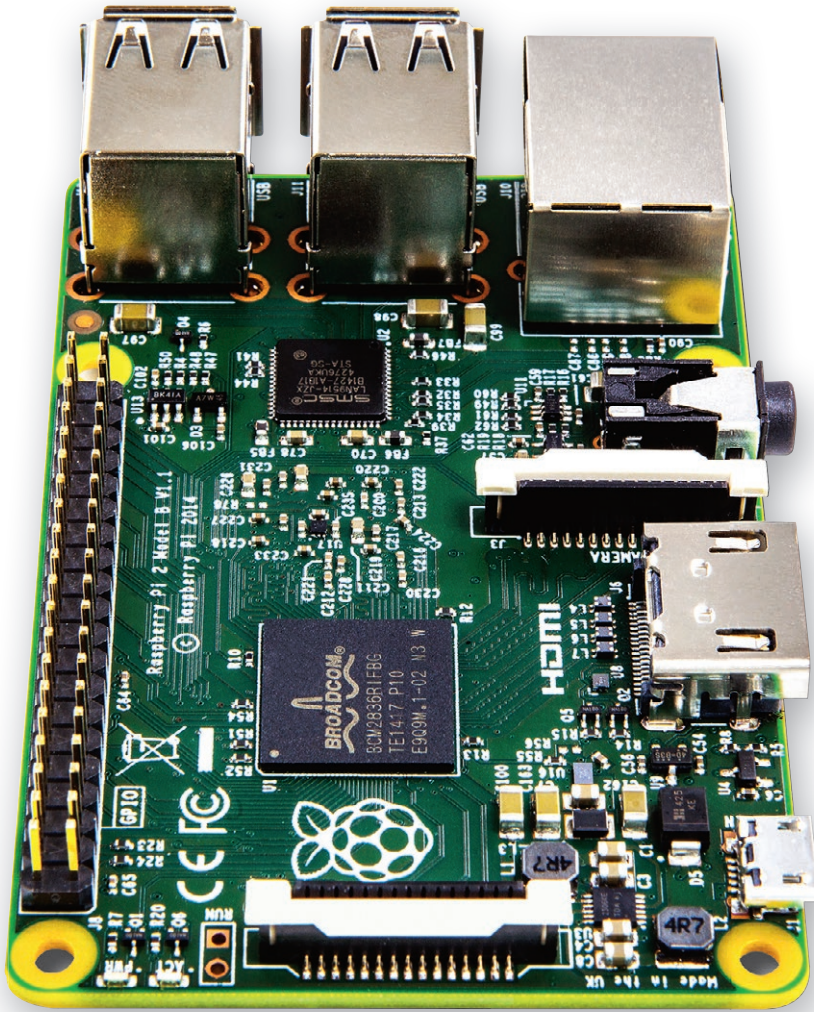
Войдя в рабочий стол, мы проверили несколько обычных приложений.

Создание нового мира в *Minecraft* заняло 42 секунды на B+ и 21 секунду на Pi 2. Загрузка IDLE 3 заняла 13 секунд на B+ и всего 4 секунды на Pi 2.

Запуск SunSpider в новом оптимизированном браузере позволил оценить производительность в реальных условиях. Выигрыш в скорости на этом наборе тестов составил 2,5 раза. Учитывая трудности многозадачности, этого и следовало ожидать. Но даже и так отдельные результаты показывали почти четырёхкратный выигрыш на этом неоптимизированном коде.

И Raspberry Pi B+, и Pi 2 содержат один и тот же графический процессор Videocore, и в наших тестах было небольшое увеличение числа кадров в секунду [FPS] для Pi 2, благодаря большему ОЗУ на плате. Последний наш тест был на скорость передачи файлов по Ethernet, в нём мы использовали **scp** для копирования 692-мегабайтного видео Big Buck Bunny на оба Pi. На B+ средняя скорость была 3,8 Мб/с, на Pi 2—4,6 Мб/с, то есть на 0,8 Мб/с больше.

Raspberry Pi Foundation выпустил обновлённый образ Raspbian, включивший ядро для ARM v7, необходимое для



➤ Может, форм-фактор и тот же, что у V+, но в Pi 2 упаковано больше мощности.

использования на новом процессоре. Но приложения, написанные для старого Raspberry Pi, полностью совместимы с Raspberry Pi 2 — полагающимся на развитые проекты, созданные со времён первого выпуска Raspberry Pi.

Raspberry Pi 2 выполнил множество пожеланий сообщества и на годы вперёд предоставил стабильную и хорошо поддерживаемую платформу для хакеров, творцов и тех, кто учится делать замечательные проекты. **LXF**

Тесты SunSpider

Тест	Pi 2	V+	Выигрыш (разы)
Всего	2760,9	8178,0	2,96
3d	550,9	1427,8	2,59
cube	157,3	473,6	3,01
morph	167,0	296,0	1,77
raytrace	226,6	658,2	2,90
access	211,9	435,9	2,06
binary-trees	27,6	69,8	2,53
fannkuch	101,5	190,1	1,87
nbody	52,8	118,7	2,25
nsieve	30,0	57,3	1,91
bitops	113,8	206,1	1,81
bits-in-byte	22,0	35,6	1,62
bitwise-and	29,1	48,2	1,66
nsieve-bits	52,8	104,1	1,97
controlflow	28,3	64,6	2,28
recursive	28,3	64,6	2,28
crypto	221,4	578,6	2,61
aes	112,4	287,6	2,56
md5	60,1	162,2	2,70
sha1	48,9	128,8	2,63
date	336,3	1269,9	3,78
format-tofte	171,5	641,9	3,74
format-xparb	164,8	628,0	3,81
math	158,4	394,5	2,49
cordic	43,3	99,9	2,31
partial-sums	78,7	215,7	2,74
spectral-norm	36,4	78,9	2,17
regex	101,9	160,6	1,58
string	1038,0	3640,0	3,51
base64	63,3	178,8	2,82
fasta	156,9	409,7	2,61
tagcloud	177,8	617,7	3,47
unpack-code	514,5	2021,6	3,93

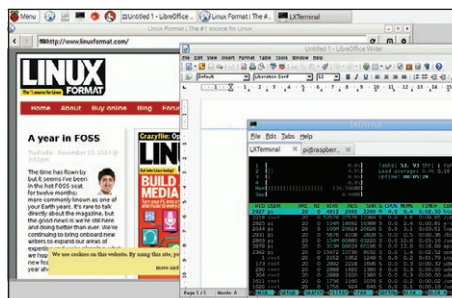
Sysbench			
Prime	74,68	509,58	6,8

Свойства навскидку



Мощный 4-ядерный ARM v7

Новый четырёхядерный ARMv7-процессор Broadcom BCM2836 с 1 ГБ ОЗУ даёт результаты тестов (см. таблицу выше) вчетверо лучше, чем старый SoC BCM2835.



Новый интерфейс Raspbian

Доступный с декабря новый, блестящий настольный Raspbian хорошо запускался и на V+, но на Pi 2 он выглядит отзывчивой системой, годной хоть и для основного ПК.

LINUX FORMAT Вердикт

Raspberry Pi 2

Разработчик: Raspberry Pi Foundation
Сайт: www.raspberrypi.org
Цена: £20

Функциональность	9/10
Производительность	10/10
Удобство в работе	10/10
Оправданность цены	10/10

➤ Почти идеальный одноплатный компьютер, связавший отменное оборудование — с притом обратно-совместимое — с живым и великодушным сообществом.

Рейтинг 10/10

OpenPi

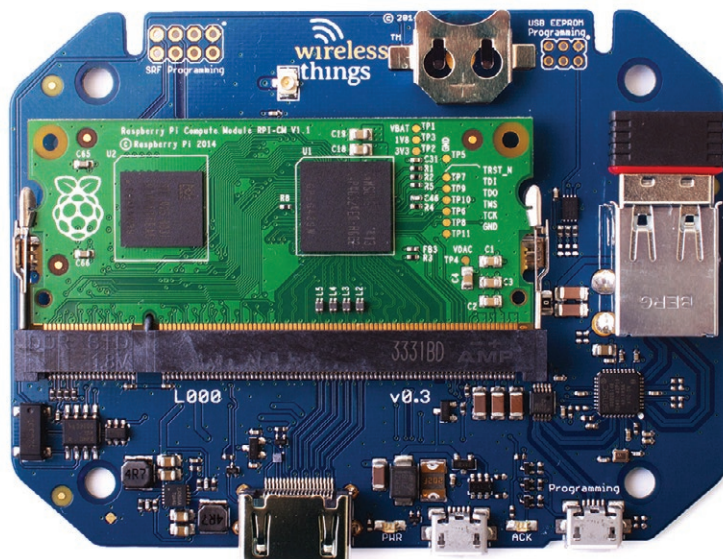
Вечно заинтригованный маленькими коробочками и схемами, Джонни Бидвелл изучает хаб Интернета Вещей для Pi, который можно изменить самим.

Вкратце

» Плата на базе Pi Compute Module с экологичной беспроводной антенной, часами и датчиком температуры. Плата и схемы доступны как Open Source, и начинающие предприниматели и изобретатели могут свободно их модифицировать.

Wireless Things — новое название Ciseco, ноттингемской компании Интернета Вещей [Internet of Things], хорошо известной своим устройством EVE Alpha. EVE — компактное устройство, удобно помещающееся поверх Raspberry Pi, подсоединяясь к контактам GPIO. Существует множество модулей, которые можно разместить на плате EVE: SRF- и RFM12B-радиодюды, часы реального времени (RTC), датчики температуры и другие. Теперь, с OpenPi, Wireless Things надеется развить эту идею дальше, и пока мы пишем эти строки, они находятся посреди процесса краудфандинга (<http://bit.ly/KickstartOpenPi>). [Сбор средств уже успешно завершён, — прим. пер.]

Имея Raspberry Pi Compute Module в качестве основы, OpenPi предоставляет два USB-порта, RTC (батарея входит в комплект), 868-195 МГц SFR приёмопередатчик (для дальнейших последовательных подключений) и датчик температуры. Всё это упаковано в опрятный пластиковый корпус с логотипом Wireless Things и отверстием для подключения HDMI и питания. В плане расширяемости вы получаете 18 (плюс 2 для питания) контактов (которые вам нужно вывести наружу самим), заголовочные файлы для программирования SRF и USB EEPROM, а также контакты для ручного подключения проводными к модулю XBee от Wireless Things. Compute Module имеет 4 ГБ памяти eMMC, на которых размещена почти стандартная установка Raspbian (для простоты последовательных подключений установлен PuTTY), но модуль можно заменить.



» OpenPi немного побольше, чем первоначальный Pi. Скошенные углы могут быть, а могут и не быть данью уважения *Battlestar Galactica*.

Спецификация

- » Чип Broadcom 2305 700 МГц, 512 МБ ОЗУ, 4 ГБ хранилища eMMC
- » ОС: Raspbian
- » 2×USB 2.0, HDMI, 2×micro USB
- » Часы: TI BQ3200
- » Датчик температуры: TI TMP1000NA
- » Передатчик: Wireless Things SRF
- » 320×213×19,3 мм, 1,34 кг

Пользователю-любителю, возможно, сошёл бы и стандартный Raspberry Pi — купить передатчики, RTC и любую другую периферию можно проще и дешевле. Компоновка будет не такой аккуратной, но вы сможете накулибинить себе подходящий контейнер из Lego, изоляты и пачки от сигарет. Базовый комплект OpenPi (включающий только Pi Compute Module с смонтированной платой OpenPi в корпусе) стоит £49 с НДС, поэтому вопрос сводится к тому, нужно ли вам всё в удобной коробке и не будет ли вам не хватать линейного аудиовыхода и Ethernet-разъёмов.

устремлениями. Об этом говорят и оптовые скидки (10% на 100 и более штук, 30% на пятнадцатые заказы), благодаря чему устройство доступнее и для образования.

Мы обнаружили, что получение доступа к RTC на нашем прототипе потребовало загрузки модуля и отправки нескольких байт по шине i2c. Датчик температуры легко опрашивается по **i2cget**, но его сырые данные требуются перепорядочить и масштабировать, чтобы прочитать правильно. Но об этих вещах следует беспокоиться после дня выпуска, это лишь вопрос написания нескольких обёрток. Сбор денег закончился 4 марта 2015 г., а выпуск запланирован на конец марта. **LXF**

Открытое железо

Уникально в OpenPi то, что все схемы этой платы открыты — вы можете скачать их в виде файлов EagleCAD. Поступая таким образом, OpenPi хочет сделать свой Compute Module более гибкой платформой, на которой можно разрабатывать проекты и продукты. Исследования производителя подсказывают, что форма не менее важна, чем функциональность, чем частично и объясняется ранее низкий интерес к Compute Module как платформе. Они заявляют, что хотя ни одна разработка не может угодить всем, использование OpenPi в качестве ориентира сильно упростит процесс разработки начинающим предпринимателям Интернета Вещей. Так что продукт реально нацелен на людей с идеями и промышленными

LINUX FORMAT Вердикт

Wireless Things OpenPi

Разработчик: Wireless Things
 Сайт: wirelessthings.net
 Цена: £49/£99 (базовый/стартовый набор)

Функциональность	8/10
Производительность	7/10
Удобство в работе	7/10
Оправданность цены	7/10

» Концептуально сильный шаг, который понравится креативным пользователям. Но домашние предпочтут уже существующие средства.

Рейтинг 7/10

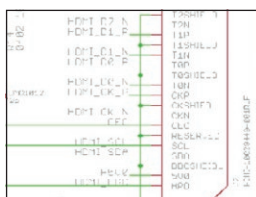


Свойства новскидку



Стартовый набор

Комплект включает 5-В зарядное устройство, кабель HDMI, беспроводную клавиатуру/тачпад и беспроводной модуль.



Открытые схемы

Любознательные могут загрузить дизайн печатных плат в EagleCAD; изобретательные могут даже изменить их.

ThinkPad Yoga 11e

При цене, уместной для ноутбука, стоит ли гибкий и прочный хромбук своих денег, интересуется **Давид Эйтельбах**.

Спецификация

- » Процессор: Intel Celeron N2930 1.83 ГГц (4-ядерный)
- » Графика: Intel HD Graphics
- » ОЗУ: 4 ГБ DDR3 (1,333 МГц)
- » Экран: 11,6 дюймов, 1366×768 HD LED
- » Память: 16 ГБ SSD (eMMC)
- » Разъёмы: HDMI 1.4, USB 3.0, USB 2.0, кард-ридер 4-в-1
- » Связь: Intel 7260 802.11ac (dual-band), Bluetooth 4.0
- » Камера: web-камера 720p HD
- » Вес: 1,5 кг
- » Габариты: 11,81×8,5×0,87 дюймов

Yoga — один из немногих на рынке хромбуков со вполне дубовым внешним видом. Созданный для жизни в суровой школьной обстановке, он может похвастаться обрезиненными краями и сверхпрочными шарнирами. Но учитывая цену в £360 с НДС, Yoga 11e более чем вдвое дороже большинства хромбуков. Достояна ли такой цены его комбинация гибкости и прочности?

Yoga 11e щеголяет той же минималистской эстетикой, что и другие ThinkPad, представляя матовые пластиковые шасси и малое количество украшений, не считая серебристых логотипов ThinkPad. При габаритах 300×216×22 мм и весе 1,5 кг, Yoga 11e немного тяжелее 13,3-дюймового Toshiba Chromebook 2. Но он всё ещё достаточно лёгок, чтобы не оттягивать сумку.

Как и у других ноутбуков серии Yoga, крышка у Yoga 11e может поворачиваться на 360° и лежать строго внизу ноутбука. Такой дизайн позволяет использовать устройство в четырёх режимах: это традиционный режим ноутбука, режима планшета (полностью перевёрнут), стоячий режим (кнопками вниз на столе) и режим палатки. В последнем случае хромбук, раскрытый на 270°, стоит «домиком» на торцах крышки и основания. Поворот экрана автоматически отключает клавиатуру.

Хотя Yoga 11e удобно держать как планшет, его польза в таком режиме ограничена. Интерфейс Chrome OS содержит густые скопления мелких кнопок, особенно в верхнем правом углу браузера, и в них сложно будет точно попасть касанием.

В отличие от большинства хромбуков, ThinkPad Yoga 11e создан, чтобы выстоять и в буйном классе, и на поле битвы. Обрезиненные края крышки защищают ноутбук от случайных падений, супер-широкая рамка минимизирует ущерб LCD-панели, а тестирование по военным спецификациям означает, что ноутбук выдерживает экстремальное давление, влажность, вибрацию, температуру и атмосферную пыль.

Если этот хромбук и упирается в ограниченность



» Хромбук, способный на лучшую гимнастику, чем русский подросток на Олимпиаде.

своего железа, то в многозадачности. Например, при работе с Google Docs одновременно с прослушиванием музыки

«Создан, чтобы выстоять и в буйном классе, и на поле битвы.»

и дюжиной других открытых вкладок аудио периодически запинаясь. Хуже того, иногда возникают полусекундные задержки после нажатия кнопок-стрелок.

Серьёзные компоненты

Приятно, что 11,6-дюймовый тачскрин Yoga 11e с разрешением 1366×768 даёт яркую и чёткую картинку. Цвета в HD-видео яркие, а текст на web-сайтах отчётливый. Однако углы обзора невелики: при перемещении более чем на 30 см в любую сторону от экрана цвета начинают размываться.

Сенсорная площадка быстро реагирует на ввод и не имеет проблем с масштабированием «щипком». Динамики на задней панели выдают на удивление хорошо сбалансированный и насыщенный звук.

Мы надеялись на выдающиеся впечатления от набора текста с ThinkPad, и не были в этом разочарованы. Клавиши «островки» имеют большой вертикальный ход и дают хорошую тактильную отдачу,

а текстурированная поверхность позволяет легко печатать касаниями. Сенсорная площадка большая и отзывчивая, прокрутка двумя пальцами делается легко, а курсор точно следует за перемещением пальца по поверхности. А вот аккумулятор подвёл: Yoga 11e проигрывает конкурирующим хромбукам. С яркостью 30%, дюжиной вкладок и потоковой музыкой Yoga продержался 6 часов 21 минуту.

Хромбук Lenovo более чем вдвое дороже многих ноутбуков на Chrome OS. Но зато он, один из немногих, по-настоящему является ноутбуком 2-в-1. Плюс, его выносливость одолеет житейские невзгоды. **LXF**

LINUX FORMAT Вердикт

Lenovo ThinkPad Yoga 11e

Разработчик: Lenovo
Сайт: www.lenovo.com
Цена: £359

Функциональность	6/10
Производительность	7/10
Удобство в работе	9/10
Оправданность цены	6/10

» Прочный и гибкий дизайн плюс отличная клавиатура. Кабы не цена...

Рейтинг 7/10

Civ: Вне Земли

Угробили свою планету? Не горюйте, советует **Расс Питт**: отправляйтесь туда, куда ещё не ступала нога человека, и отравите целую вселенную новых планет.

Вкратце

» Новый научно-фантастический подход к серии *Civilization*. Отправляйтесь в экспедицию и найдите дом вне Земли, исследуйте и колонизируйте новые планеты и создавайте новую цивилизацию в космосе.

Civilization: *Beyond Earth* [Цивилизация: Вне Земли] начинается вполне научно-фантастическим предположением «что, если...». Что если взять *Civilization*, классическую пошаговую стратегию, и сделать одну из её знаковых концовок началом новой игры?

Beyond Earth содержит много нового: юниты, победы, полностью новое дерево технологий (на самом деле, паутину), новые лидеры, новые цивилизации, и полезные вещи внутри тоже совершенно новые. Но важнее то, что впечатления от открытия, наблюдения за посадкой колониального корабля в абсолютную темноту карты и затем первого шага в инопланетный мир — такие же, как от игры в *Civ 5*.

Если вы ищете в *Beyond Earth* прямое продолжение или обновление игры 1999 г., то будете дико разочарованы. Хотя здесь есть знакомые детали повествования, и сюда закралась кое-какие узнаваемые аспекты *Alpha Centauri* [Альфы Центавра], *Beyond Earth* во многом самобытна.

Это заметнее всего при взаимодействии с аборигенами планеты. Вместо варваров *Civilization*, в *Beyond Earth* есть ряд инопланетных форм, одна агрессивнее другой. Ещё одно большое нововведение — орбитальный слой. Вы можете строить и запускать спутники, и они будут давать преимущества конкретным зонам.

А теперь плохая новость: легко заметить, что *Beyond Earth* — всего лишь расширение *Civ 5*, хотя и более космическое, чем прежние. Для фанатов *Civ 5* вроде меня это неоднозначное предложение. Кому нравится *Civ 5*, то понравится и ещё одна *Civ 5*, но трудно отрицать, что при всей нашей любви мы ожидали большего, чем *Civ 5* в шкуре научной фантастики.

Многочисленные черты сходства *Beyond Earth* с *Civ 5*, к сожалению, отвлекают от того факта, что это определённо новая



» В *Beyond Earth* есть пять вариантов победы: один из них — Освобождение, при котором вы строите портал Освобождения, чтобы вернуться на Землю и покорять её.

игра, и если научиться не замечать эти сходства, то нас захлестнёт новизна и радость игры в *Civ* будущего, и игра завладеет нами раньше, чем мы это заметим.

Не просто Civ 5 в космосе

Выбрав Бразильскую цивилизацию, мы придерживались философии Праведности [Purity], но вынуждены были нащупывать свой путь по паутине развития технологий и в итоге проиграли, не заметив, как один из наших противников был близок к победе. Но начали заново, на сей раз как Славянская Федерация. Избрав Превосходство [Supremacy] и немного поразведав, мы стремились завершить игру Контакт с другой расой, построив достаточно сильную цивилизацию, чтобы как минимум покорить мир, если уж ничего другого не выйдет. *Beyond Earth* предлагает 5 условий победы, хотя два из них отличаются лишь тем, какая философия к ним приводит. Контакт предполагает обнаружение инопланетного сигнала и раскрытие тайны Прародителей — древней расы пришельцев.

Господство [Domination] соответствует своему названию, давая вам победить, если вы захватите столицы всех вражеских цивилизаций. Земля Обетованная [Promised Land] и Освобождение [Emancipation] — две стороны одной медали. Надо развить технологии, чтобы открыть портал Освобождения, или Исхода обратно на Землю, и привести людей туда, оставив позади и освобождение, и господство. Трансцендентность [Transcendence] — это победа для

последователей Гармонии [Harmony]. Она предполагает развитие технологии вплоть до создания «цветка разума [mind flower]», который сольёт ваше сознание с сознанием планеты.

Во время третьего прохождения мы решили побеждать без выстрелов. Мы выбрали Франко-Иберийскую цивилизацию и сосредоточились на философии Гармонии. Войны так и не случилось. Хотя нам и пришлось убить горстку агрессивных инопланетян, мы завоевали господство с помощью науки и торговли, без оружия. И когда наш цветок разума расцвёл, мы наконец прочувствовали и поняли всё, что предлагает *Beyond Earth*.

Тем-то *Beyond Earth* и хороша: она даёт игру с традициями и управлением *Civilization*, однако новую и способную удивлять в зачастую неожиданных местах. **LXF**

LINUX FORMAT Вердикт

Civilization: Beyond Earth

Разработчик: Aspyr
Сайт: GameAgent.com
Цена: £ 30

Игровой процесс	9/10
Графика	8/10
Продолжительность	8/10
Оправданность цены	8/10

» Из-за родства с *Civilization 5* игра кажется знакомой, однако полна интересных сюрпризов.

Рейтинг **8/10**

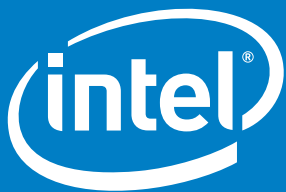
Спецификация

Минимальные требования:

- » ОС: Steam OS, Ubuntu 14.04
- » Процессор: Intel Core i3 или AMD A10
- » Память: 4 ГБ
- » Графика: Nvidia Geforce 260
- » На диске: 8 ГБ

» Цветок разума. Обеспечьте победу, используя гнусную мощь планетарного сознания.

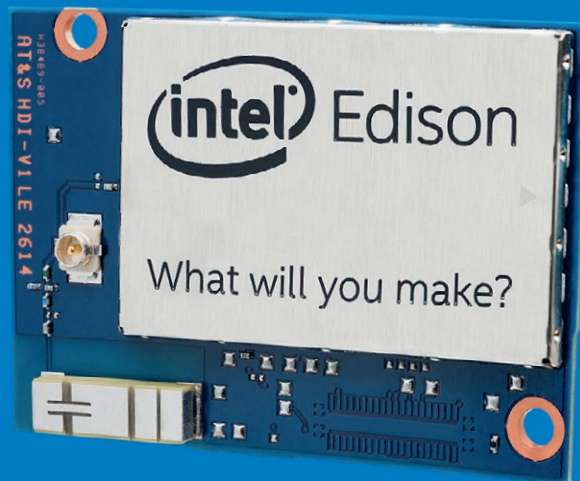




Intel® Edison

Впервые в России [официально].

Оформи предзаказ на
www.linuxcenter.ru



Выиграй поездку*
на Geek Picnic
[19–21 июня] или
Intel IoT Roadshow
[27–28 июня]
в Санкт-Петербурге.

Москва: +7 (499) 271-49-54
Санкт-Петербург: +7 (812) 309-06-86
E-mail: info@linuxcenter.ru



С полными условиями конкурса
можно ознакомиться на
www.linuxcenter.ru/intel/edison



* в призовом фонде 6 поездок



ANDROID МУЖАЕТ

Console OS: Android на ПК

«Телефонный» дистрибутив стал настольным промышленного уровня.

Теперь Android может стать основной операционной системой на вашем ПК! Если вам необходимо запускать Android на ноутбуке или настольном компьютере, а Android x86 по какой-то причине для этой цели не подходит, рекомендуем обратить внимание на проект Console OS.

Разработчики характеризуют свое решение как «дистрибутив Android промышленного уровня для десктопов», предназначенный для использования в качестве основной операционной системы на ПК. Console OS может быть установлен на рабочие станции, ноутбуки, планшеты, другие аппаратные средства платформы x86; также поддерживается загрузка системы с внешнего USB-носителя. Изначально продукт распространялся как коммерческий дистрибутив Android, теперь же разработчики выпустили бесплатную версию Console OS — «standard edition». В ней, к сожалению, отсутствуют некоторые дополнительные функции, предлагаемые Console OS Pro, в том числе

возможность переключения в реальном времени между Windows и Android.

В основу Console OS «standard edition» положена система Android Lollipop от Google. Среди конкурентных преимуществ Console OS — поддержка двойной загрузки с Windows 8.1, полная совместимость с Android KitKat, возможность параллельного запуска двух приложений Android, дружественный «настольный» графический интерфейс с привычными кнопками и меню. Не позабыты и игры: в настоящее время Console OS поддерживает OpenGL ES 3.1, а к середине 2015 г. обещана полная реализация стека OpenGL 4. В планах разработчиков также реализация поддержки Secure Boot и dm-verify.

Для загрузки Console OS «standard edition» или получения дополнительной информации посетите сайт проекта — ConsoleOS.com. Базовая версия для загрузки предлагается бесплатно, однако требуется регистрация.

	Console OS with Android™ inside	android-x86.org	Bluewin's	Andy (Beta)	Android on Organ
Type	Native Android w/ Dual Boot on Win 8.1 (Pathless to Install)	Native Android w/ Dual Boot & Hypervisor (NOT compatible with Windows 8.1 Secure Boot)	Emulator (JIT Translation Via OVMF) Mostly Partially Slow	Emulator (JIT Virtual Machine Windows Drops It Down) Emulator Interface	Native Android for Only. Not Applicable
Automatically Dual Boots Windows 8.1 (Pathless to Install)	YES	No (As late as ten seconds to toggle with Windows*)	No	No	No
Full Android KitKat Compatibility	YES	No (Bugs)	No (Old Android)	No (Old Android)	Partial (Some)
Run Top-Android Apps Side by Side	YES	No	No*	No*	No
Designed for Consumers & OEMs	YES	No	If They Say So...	Perhaps - Unlicensed	No
OpenGL ES 3.1 & 3.1.1	YES	No	No	No	Partial 2.1
Full OpenGL 4.1 w/ Minimum Query	Planned Mid-2015	No	No	No	Yes
IronPython™	YES	No	No	No	No
Light Google Play App Support with Native App Store	YES	No (Unlicensed)	Yes	No (Unlicensed)	No
ARM & x86 Native App Compatibility	YES	No (Unlicensed) Same as above	No	No (Unlicensed) Same as above	No
Toggle from our UI to Pure Android UI	YES	N/A (Stock Android)	No	N/A (Stock Android)	N/A (Stock)
Desktop-friendly UI, Menu, Buttons	YES	No	No	No	No

» Все, что сделало мобильную платформу удивительной, переносится на ваш ПК.

ДИВЕРСИФИКАЦИЯ

Время меряет Windows

ASUS в своих часах откажется от Android Wear в пользу Windows.

Для будущих моделей своих «умных» часов ASUS подыскивает операционную систему, отличную от Android Wear, называя время работы от батареи главной проблемой платформы Google. «Мы продолжим сотрудничество с Google в части Android Wear, однако у нас будут и другие часы, не связанные с Android Wear и обеспечивающие более длительное время работы между перезарядками», приводит издание Focus Taiwan слова CEO ASUS Джерри Шена [Jerry Shen].

В настоящее время ASUS предлагает одну модель часов на Android Wear — ZenWatch, и планирует выпустить модель второго поколения в III квартале текущего года. Известно, что время работы от одного заряда у большинства часов на Android Wear составляет один-два дня. Для сравнения, в своих первых non-Wear часах ASUS рассчитывает обеспечить 7 дней работы на одной зарядке. Г-н Шен упомянул не только другую

операционную систему, но и более простой чипсет от MediaTek, который пока еще находится в разработке.

Г-н Шен не стал углубляться в специфические особенности платформы, однако имеющаяся разрозненная информация свидетельствует

«Производители не изобрели „формулу успеха“ для умных часов.»

о сотрудничестве ASUS и Microsoft, направленном на создание часов, работающих под управлением специализированной версии Windows. В этом есть определенный смысл, учитывая желание Microsoft лицензировать свою технологию Band для других производителей «умных» часов;

однако в настоящее время Microsoft Band обеспечивает не более 48 часов автономной работы. Компания Pebble, чьи часы способны работать между перезарядками целую неделю, также имеет планы сотрудничества с другими производителями оборудования, и уже сделала свои часы совместимыми со всеми приложениями Android Wear.

Снова приходится констатировать: производители пока что не изобрели «формулу успеха» для умных часов. На Android Wear возлагались большие надежды, которые, тем не менее, не переросли в успешные продажи. ASUS находится в числе тех компаний, которые предпочитают «не хранить все яйца в одной корзине» и, наряду с LG, Samsung и (по некоторым данным) HTC, хочет подстраховаться, сделав ставки на разные платформы.

НОВАЯ ПОПЫТКА

Городские — не такие

LG хочет притянуть покупателей к часам на Android Wear.

Компания LG 15 февраля анонсировала умные часы G Watch Urbane — получившее цельнометаллический корпус продолжение модели G Watch R, той самой, которую обозреватели уже назвали «лучшей, с эстетической точки зрения, моделью на Android Wear». И если G Watch R внешне напоминали спортивный хронометр, то с G Watch Urbane на руке их владельцу не стыдно будет посетить самый изысканный светский раут.

Да, это не Rolex и не Patek Philippe, но на фоне других часов на Android Wear, G Watch Urbane выглядят просто шикарно. Этому способствуют и более тонкий, чем у предшественника, корпус, и традиционный круглый дисплей; а вот, например, ZenWatch от ASUS со своими прямоугольными формами смотрятся как типичный «гаджет». В продажу новинка поступит в золотом и серебряном корпусе, со сменным кожаным 22-мм ремешком. Кстати, короткие проушины, к которым крепится

ремешок — то самое дизайнерское решение, которое позволяет всю линейку G называть именно часами, а не «гаджетами». Функциональность и цену новинки LG озвучила в марте на конференции Mobile World Congress в Барселоне. Учитывая соответствующее оформление, ценник из разряда премиум неудивителен: в интернет-магазине Exrandsys версии в золотой и серебряной отделке предлагаются за €359,99.

Ключевые спецификации новинки аналогичны G Watch R. Те же самые 1,3-дюймовый дисплей 320×320 P-OLED, процессор Snapdragon 400 1,2 ГГц и батарея ёмкостью 410 мА·ч. В модели G Watch R эти компоненты обеспечивают чёткое цветное отображение при солнечном свете, яркое чёрно-белое — в режиме dim/ambient, и одно из лучших среди устройств на Android Wear время автономной работы.

Не стоит считать G Watch Urbane прямым ответом на скорое и неизбежное появление Apple



➤ Предшественник, G Watch R, смахивает на секундомер, а G Watch Urbane щеголяет элегантностью.

Watch: пользователи Android или iOS, как правило, безразличны к ситуации с продажами в конкурирующем лагере. Новая модель — это еще одна попытка производителей устройств на Android Wear, продажи которых, за исключением разве что Moto 360, идут довольно вяло, привлечь интерес потребителей.

КОНВЕРГЕНЦИЯ

Уведомления на всех

Pebble обеспечила совместимость с Android Wear.

Платформа производителя «умных» часов Pebble теперь совместима с приложениями для операционной системы Google Android Wear и позволяет владельцам видеть и реагировать на поступающие с их запястий уведомления. С декабря минувшего года данная функциональность находилась в стадии бета-тестирования, а со второй декады февраля стала общедоступной.

ответы на уведомления, которые упрощают работу с ними. Чтобы воспользоваться интеграцией с Android Wear, владельцам Pebble необходимо обновить прошивку своих часов и загрузить последнюю версию приложения Pebble для Android.

Стратегия Pebble на рынке становится всё более конкурентоспособной. По данным компании Canalys, непрерывный поток обновлений программного обеспечения, большое количество

приложений в App Store компании и снижение цен способствовали поддержанию высокого уровня продаж устройств Pebble во второй половине минувшего года.

Тактика реализации в своей платформе поддержки приложений для Android может привести к непредсказуемым результатам, примером чего является компания BlackBerry и её смартфоны. Однако позиция Pebble более выгодна за счёт того, что приложения для «умных» часов пока не так совершенны, как их аналоги для смартфонов, и пока востребованы на меньшем количестве базовых платформ. Последующие 12 месяцев станут для рынка «умных» часов судьбоносными: выход в апреле Apple Watch является наиболее ожидаемым событием в этом



➤ Уведомления, раньше доступные только владельцам часов с Google OS, теперь воспроизводятся и на продуктах Pebble.

сегменте. «Apple Watch способны стать тем трамплином, который позволит рынку носимой электроники взять новый рубеж. Успех Apple создаст волну, на гребне которой поднимется весь рынок», уверяет глава исследовательского отдела компании CCS Insight Бен Вуд [Ben Wood].

В противном случае, т.е. если продажи Apple Watch не оправдают ожиданий, это больно ударит по всему рынку «умных» часов. Однако в нынешнем году нам ещё предстоит увидеть, что приготовила Pebble: компания работает над новыми продуктами, и её программная платформа остаётся актуальной. **LXF**

«Последующие 12 месяцев станут для „умных часов“ судьбоносными.»

Как известно, Уведомления являются одним из основных инструментов Android Wear, и функциональность, ранее доступная только владельцам часов с Google OS, теперь может быть воспроизведена и на продуктах Pebble.

Подготовленные Pebble видеоролики демонстрируют, как пользователи отвечают на сообщения Google Hangout и подтверждают денежные транзакции через Square Cash. В ходе бета-тестирования Pebble также добавила predetermined

Сравнение

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

Языки скриптов

Ричард Смедли выходит за пределы *Bash*, выясняя, какие языки скриптов соответствуют потребностям и желаниям системных администраторов Linux.



Про наш тест...

Говорят, сравнения неправомерны изначально. Что особенно верно относительно языков программирования, где личные пристрастия и поддержка на местах имеют как минимум не меньшую значимость, чем такие критерии, как скорость и уровень поддержки для разных парадигм. Учитывая это, мы представляем вам набор фактов, общественных мнений и наших собственных предубеждений, и это — лишь основа для дальнейших исследований. Суть пользы языка скриптов для сисадмина заключается не в том, легко ли он позволяет решать проблемы, а в том, сколько решений уже написано и готово для скачивания и внедрения, и, желательно, хорошо задокументировано.

Мы попытались поработать с рядом версий, установленных в типовой сети, но выбрали Python 3. В остальном мы старались остаться в контексте всего того, с чем вы обычно сталкиваетесь в своей сети.

```
memory
));
# memory.h: ARM has a custom one

sub build_types {
  my $mods = "(?x: \n" . join("| \n ", @modifierList) . "\n)";
  my $all = "(?x: \n" . join("| \n ", @typeList) . "\n)";
  my $allWithAttr = "(?x: \n" . join("| \n ", @typeListWithAttr) . "\n)";
  $Modifier = qr{(?:$Attribute|$Sparse|$mods)};
  $NonptrType = qr{
    (?:$Modifier\s+|const\s+)*
    (?
      (?:typeof| typeof )\s*\([^\)]*\)|
```

Все сисадмины любят экономящие время горячие клавиши и переносят с одной работы на другую накопленный набор скриптов, обогащая его на каждом новом месте. Каждый новый администратор задается вопросом: какой язык ему изучать? (Это второй по популярности вопрос после «Где здесь дают кофе?») Ветераны языковых войн знают, что на вопрос о лучшем языке вряд ли есть простой или однозначный ответ, однако мы сочли правильным сравнить самые удобные варианты, чтобы немного облегчить вашу жизнь в Linux.

Большинство языков скриптов существуют гораздо дольше, чем вы думали:

скажем, newLISP появился на рабочей станции Sun-4 в 1991 г. Они постоянно что-то брали друг у друга и из других источников, и накопили солидный багаж устаревших библиотек и наработок. Например, регулярные выражения Perl сейчас находятся буквально повсюду, и иной раз они лучше реализованы на стороне. И что же нам важнее? Скорость работы скрипта или скорость его написания? В большинстве

случаев, последнее. Когда все готово к работе, нужна поддержка со стороны библиотек или модулей, чтобы распространить язык на все области вашей работы, и со стороны достаточно крупного сообщества, чтобы поддержать язык, помочь ему не отставать от последних тенденций и даже внедрять в него инновации. Какой же язык скриптов стоит изучить, чтобы вам легче жилось в Linux в этом году?

Наша подборка

- » Bash
- » Perl 5
- » Python
- » Ruby
- » newLISP

«Вопрос, которым задается каждый новый администратор — какой язык изучать.»

Процесс изучения

Онлайн-ресурсы, книги и хорошие люди.

Главные вопросы: насколько прост выбранный язык? Являются ли обучающие ресурсы хотя бы адекватными? И даже если на оба ответа утвердительно, все же вам нужна поддержка доброжелательного сообщества, способного помочь быстро наваять нечто полезное и не дать угаснуть вашему энтузиазму при неизбежных проблемах.

Чтобы в каждом из языков создать скрипт резервного копирования и тестовые скрипты, мы начали с поиска Stack Overflow [Переполнение стека]. Но скачивание случайного кода означает плохую совместимость скриптов Posix (чистая Bourne Shell), современного *Bash* и старого кода, который периодически дает сбой. К счастью, www.shellcheck.net — отличный инструмент для проверки корректности скриптов, и он их исправляет, обучая вас правильным решениям. Всесторонний (возможно, даже слишком) Advanced Bash Scripting Guide (www.tldp.org/LDP/abs/html) от Linux Document Project также превосходит: с его помощью освоитесь вы быстро.

Онлайн- и встроенная документация Perl вошла в легенды, но мы для начала прошлись по упражнениям из классики: книги от O'Reilly для администраторов,

Running Linux [Запускаем Linux], а потом перепрыгнули через пару десятков лет на недавнюю *Perl One-Liners* [Скрипты-однострочники на Perl] Петериса Круминьша [Peteris Krumins] от No Starch. Те, кто не хочет иметь дело с книгой, могут обратиться к <http://perlmonks.org>, источнику коллективной мудрости сообщества.

Недавние попытки увлечь молодежь учебой через Code Club (www.codingclub.co.uk), а остальных — через образовательные спринты и открытые дни хакерства RuConUK, показали, что Python очень легко воспринимается всеми. Но старые рекомендации (скажем, изобилие способов запуска подпроцессов, уцелевших по причинам совместимости) означают необходимость тщательного изучения материала, и это — еще один хороший повод начать с Python 3, а не Python 2. На www.python.org/about/gettingstarted имеется солидный список бесплатных руководств и ресурсов.

Ruby — тоже не сложная для начинающих оболочка, и до Rails ей лучше всего удавалось справляться с приложениями командной строки. Книга Дэвида Б. Коупленда [David B. Copeland] *Build Awesome Command Line Applications in Ruby* [Создание чудных

First things first

So, you've never programmed before. As we go through this tutorial, I will attempt to teach you how to program. There really is only one way to learn to program. You must read code and write code (as computer programs are often called). I'm going to show you lots of code. You should type in code that I show you to see what happens. Play around with it and make changes. The worst that can happen is that it won't work. When I type in code it will be formatted like this:

```
##Python is easy to learn
print("Hello, World!")
```

» От MOOC до книжных магазинов, ресурсы для изучения Python имеются повсюду.

приложений командной строки в Ruby] сэкономит вам долгие часы блуждания по онлайн-документации; ну, а мы справились с нашими тестовыми скриптами благодаря парочке сетевых руководств.

И, наконец, newLISP: это сущее испытание для программистов, воспитанных на других семействах языков, но вас поразит, как он справляется с задачами всего лишь посредством списков, функций и символов. Ради нашего скрипта резервного копирования мы нырнули за фрагментами кода на <http://newlisp.org>, и наши труды вознаградила лаконичный и мощный код, который было проще читать, чем его не менее компактный конкурент Perl.

Вердикт

Python ★★★★★
 Ruby ★★★★★
 Bash ★★★★★
 newLISP ★★★★★
 Perl 5 ★★★★★

» Python и Ruby проще изучать благодаря хорошей документации и доброму сообществу.

Версии и совместимость

Как одолеть проблему неправильной версии.

Ата ли у меня версия? Начнем с *Bash*. Любой современный дистрибутив Linux идет с версией, которая запустит и ваши, и чьи угодно скрипты. *Bash 4*, со своими ассоциативными

массивами, сопроцессами (коммуникацией между двумя параллельными процессами) и рекурсивным соответствием через globbing (используя ** для расширения имен файлов) появился 6 лет

назад. *Bash 4.2* мало что добавил, и ему уже 4 года, а изменения в *Bash 4.3* мелкие.

Perl поставляется большинством дистрибутивов. Последняя версия — 5.20 (скоро выйдет 5.22), а многие стабильные дистрибутивы идут с 5.18. Но разница версий лишь в незначительных улучшениях.

Переход с Python 2 на 3 — пока опрометчивый шаг. Запускайте Python 3, а не удастся — загляните в документацию. Для Python 3 наша точка отсчета — Python 3.3, а Python 3.4 синтаксиса не менял.

Изменения версий Ruby вызвали достаточно проблем, чтобы породить решение-обезболиватель: *rvm* дает запускать несколько версий Ruby, а *bundle* отслеживает нужные для каждого скрипта фрагменты-жемчужины, называемые здесь gems.

Преимущества newLISP — стабильность и отсутствие сторонних скриптов. Но мы не можем гарантировать работу всех скриптов на самых последних версиях.

```
#- Print the 'Info Bar'
#-----
print "\n${l_hlt}" . " " "${l_columns}" . "${l_rst}\r"
. "${l_hlt}" "${l_int}BRANCH:${l_rst}${l_hlt}" "${l_git_branch}" "${l_git_sha1}"${l_msg}
} "${l_rst}\n$ ";
EOF
}

export COLUMNS
PS1="${PS1}\${Func_gitCheck}"
Func_gitCheck
richard@luggable:~/work$ bash --version
bash --version
GNU bash, version 4.3.11(1)-release (i686-pc-linux-gnu)
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Func_gitCheck
richard@luggable:~/work$
```

» Поскольку оболочка Unix насчитывает уже не одно десятилетие, вы увидите, что недавние версии *Bash* содержат некоторые неожиданные изменения в синтаксисе.

Вердикт

Bash ★★★★★
 newLISP ★★★★★
 Perl 5 ★★★★★
 Ruby ★★★★★
 Python ★★★★★

» Работа с версиями Ruby очень неплоха, но уж лучше отсутствие проблем в Bash.

Скрипты специально для Web

Заставьте свои скрипты администрирования работать с HTTP.

Большая часть жизни системного администратора переехала в Сеть, и вам, естественно, требуется язык скриптов, который от этого не отстал. Мы проверили простоту написания собственного кода и наличие готовых решений для выполнения разных задач, от web-интерфейса до системной статистики.

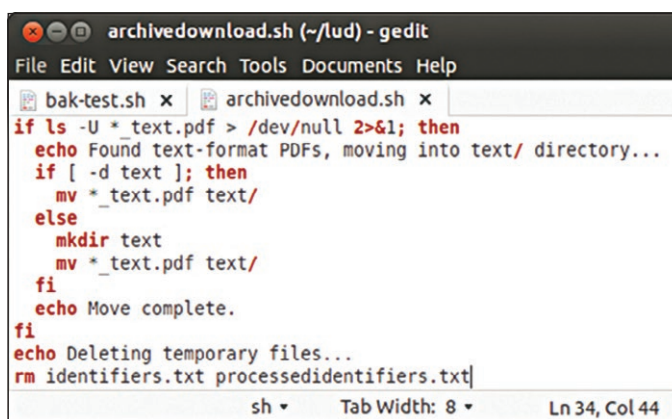
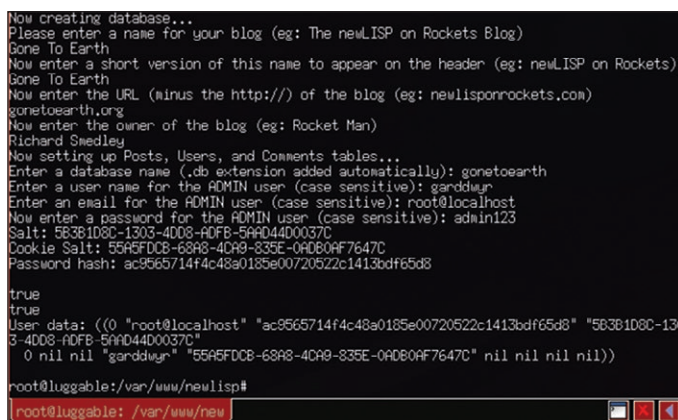
Весьма заметна в этих языках разница в выразительности и стиле при достижении одних и тех же результатов. Однако это, еще раз повторим, для многих администраторов определяется скорее персональными пристрастиями и наличием поддержки на месте. Ruby отличается быстротой и приятностью в работе; Python

«кажется правильным», вероятно, благодаря своей человеко-читаемости; newLISP удивительно мощный. Однако все эти наблюдения остаются субъективными клише при отсутствии среды с хорошей поддержкой и управляемостью, в которой используется и создается код для вашей собственной сети.

Bash ★★☆☆☆☆

Хотя к *Bash* вряд ли кто-то обратится в первую очередь, выбирая язык web-программирования, все же неплохо знать, что если ваш сервер не предоставляет вам желаемый вариант, вас может выручить *bashlib*. Это скрипт оболочки, который делает программирование CGI [графический интерфейс] в оболочке *Bash* чуть более терпимым.

В вашем скрипте будет множество команд **echo**, прославивших ваши действующие команды. По причинам безопасности мы не рекомендовали бы запускать скрипты в открытом Интернете, но стоит иметь в виду, что *Bash* хорошо работает как язык-прототип. Очень легко нашпиговать текстовый файл комментариями, описывающими требуемую структуру в общих чертах, а затем заполнять пробелы — тестируя фрагменты кода интерактивно и отправляя их в www.shellcheck.net, чтобы по ходу работы проверить код. Доказательство концепции вы получите очень быстро.



newLISP ★★★★★☆

Code Patterns от создателя newLISP Лутца Мюллера [Lutz Mueller] имеется на сайте www.newlisp.org, и там есть главы, посвященные HTTPD и CGI, а также коммуникациям TCP/IP и UDP; добавить туда раздел об управлении приложениями — и у вас будет все необходимое для начала работы.

Встроенная работа в сети newLISP и простой синтаксис (или его полное отсутствие) поразительно упрощает задачу создания HTML страниц по результатам, например, скриптов мониторинга. В качестве готовой среды есть newLISP on Rockets, которая использует Bootstrap, jQuery и SQLite, объединяя быструю разработку приложений с хорошей производительностью.

newLISP on Rockets предлагает несколько функций, от (convert-json-to-list) через (twitter-search) до (display-post-box), которые помогут вам добавить сетевые функции. Нас это очень впечатлило; но смущает скромный размер сообщества и неровная скорость разработки.

Поддержка сообщества

Хватает ли сообщества для реальной поддержки в работе?

ДevOps, облачная разработка, тестирование и непрерывная интеграция — требования к сисадминам постоянно меняются и развиваются, но требование постоянно изучать нечто новое остается вечным. Все до какой-то степени используют *Bash*, однако вы должны знать *Bash* плюс что-то еще.

Perl — традиционный всесторонний инструмент администраторов Unix 1980-х и 1990-х, который в последнее десятилетие постепенно уступает место Python, а затем Ruby. Все, кто приступал к работе

в 1990-е или раньше, прекрасно с ним поладят, и найти того, кто поможет вам с вашим скриптом, не составит труда.

Однако мир не стоит на месте, и многие технические компании стали использовать в качестве стандарта Python, который, например, весьма интенсивно употребляется в Google. Многие программы, необходимые для работы современного системного администратора, основаны на Python, хотя то же можно сказать и о Ruby.

Преимущество Ruby в том, что он является основой Chef и Puppet, а также Vagrant

и Travis CI, то есть некоторое знакомство с ним не помешает везде, где они используются. Сетевые интерфейсы и инструменты тестирования, написанные на Ruby, способствовали популярности этого языка во многих начинающих web-компаниях.

У newLISP намного меньше сообщество и мало готовых решений, и вам, возможно, не знаком ни один его пользователь. Онлайн-сообщества отчасти могут восполнить этот недостаток, но вам придется искать, кто поддержит ваши инструменты, если вы покинете компанию.

Вердикт

- Ruby ★★★★★
- Perl 5 ★★★★★
- Python ★★★★★
- Bash ★★★★★
- newLISP ★★☆☆☆☆

» Ruby привлекает к себе внимание благодаря отличным программам DevOps.

Perl 5 ★★★★★★

Perl был первым языком web-скриптов CGI, но более или менее идет в ногу со временем. В нем, понятно, есть библиотеки, и хватает примеров для изучения, но доминирующего решения нет, и стоит быть осторожным в выборе.

Catalyst, Dancer и Mojolicious — хорошие среды web-приложений. Еще более вероятно, что вы найдете все необходимое в CPAN. Чтобы справиться с определенными задачами, например, по нахождению данных XML, конвертированию данных в PDF и их индексированию на web-странице, можно соединить несколько библиотек — многие из них уже собраны в дистрибутивах.

Традиционный графический интерфейс Perl по-прежнему доступен, и, несмотря на лучшую альтернативу в плане производительности, отображенную посредством PSGI, вы, возможно, решите, что **use CGI**; — это все, что вам нужно для активации своего скрипта в Сети. И помните: «всегда есть несколько вариантов сделать это».

```

: %s' % content_id, content)
    else:
        filename, _, _ = related_attachment
        content = re.sub(r'(?<cid:)%s' % re.escape(filename),
s' % filename, content)
        self.alternatives[i] = (content, mimetype)

    return super(EmailMultiRelated, self).create_alternatives(msg)

def _create_related_attachments(self, msg):
    encoding = self.encoding v settings.DEFAULT_CHARSET
    if self.related_attachments:
        body_msg = msg
        msg = SafeMIME multipart(_subtype=self.related_subtype, encoding=
        if self.body:
            msg.attach(body_msg)
        for related_attachment in self.related_attachments:
            if isinstance(related_attachment, MIMEBase):
                msg.attach(related_attachment)
            else:

```

```

if ($q-param()) {
    # Parameters are defined, therefore the form has been submitted
    display_results($q);
} else {
    # We're here for the first time, display the form
    output_form($q);
}

# Output footer and end html
output_end($q);

exit 0;

#-----

sub output_top {
    my ($q) = @_;
    print $q-start html(
        -title => 'Back-up selection'

```

Python ★★★★★★

Python Web-Server Gateway Interface (WSGI), который был определен в PEP 333, абстрагируется от web-интерфейса, а библиотеки WSGI занимаются управлением сессиями, аутентификацией и почти всеми проблемами, которые хочется решить с помощью промежуточного ПО. Кроме того, в Python множество полноценных web-сред, таких, как Django, TurboGears и Pylons. Так же, как в Rails, для ряда задач будет лучше взять некоторые web-функции из готового скрипта. Но движок шаблонов Python не даст вам учинить мешанину из HTML и Python.

Python обладает целым рядом других преимуществ, от облака Google App Engine с собственным интерпретатором Python, который работает с любой средой web-приложений, совместимой с WSGI, с целью тестирования масштабируемых приложений, до поддержки аккуратного стиля метапрограммирования.

Ruby ★★★★★★

Ни на минуту не думайте, что Rails — панацея для проблем системных администраторов. Это не так. А Sinatra, хотя определенно облегчает написание какого угодно сетевого приложения в Ruby, для многих задач будет перебором. Rails отлично справляется с задачей наскоро слепить код, однако просто не впускается во всё волшебство создания кода.

Ruby идеально превращает любой скрипт в web-скрипт, благодаря gems, написанным умными людьми, которые принимали здравые решения. Например, включение web-интерфейса в наш скрипт резервного копирования было истинным удовольствием, но нас это сильно отвлекло, потому что мы заигрались с посторонними gems, например, для экспорта отчетов в электронные таблицы Google. Такие инструменты, как *lanc*, которые генерируют статический HTML из HAML, и ряд образцовых отчетов дополняют выразительность языка, а уж задача расширения функциональности скрипта решается мигом.

```

lib:
development,sqlite3 seeds,rb

lib:
assets capistrano tasks

log:
development.log

public:
04.html 422.html 500.html favicon.ico robots.txt

test:
controllers fixtures helpers integration mailers models test_helper.rb

tmp:
cache pids sessions sockets

vendor:
assets
richard@luggable:~/work/code/ruby/rails/crash/code/blog$
richard@luggable:~/work/co

```

Программируемость

Большие скрипты требуют иной парадигмы программирования.

Еще до достижения 1000 строк кода скрипты *Bash* становятся неуправляемыми. Хотя по природе он процедурный, имеются попытки создания объектно-ориентированного (ОО) *Bash*. Не станем такое рекомендовать: по-нашему, лучше использовать модули. Функциональное программирование (FP) в *Bash* (<http://bit.ly/BashFunsh>) тоже непрактично.

Perl, с упором на ОО, понравится далеко не всем, но с работой справится. Благодаря полнофункциональным замыканиям

его можно сделать FP, несмотря на проблемы с синтаксисом. Правда, без особой красоты: тут придется подождать Perl 6.

Python хорош и как императивный, и как ОО, и справляется с FP. Функции — объекты классные, но иных возможностей не хватает, хотя Python ладит со списками. Mochi, язык FP (<http://bit.ly/FPMochi>), использует интерпретатор на Python 3.

Ruby создан как чистый ОО-язык; вероятно, он лучший со времен Smalltalk. Вдобавок его можно уломать на поддержку

функционального стиля программирования. Но для этого надо настолько отдалиться от хорошей обычной практики, что это кажется совершенно другим языком.

И у нас остается newLISP, элегантный и мощный язык со всеми нужными функциями у вас под рукой. newLISP использует реализацию псевдо-ОО в форме функционально-объектно-ориентированного программирования (FOOP), однако это не означает, что он может извлекать из нее настоящее ОО-программирование.

Вердикт

- Python ★★★★★★
- Ruby ★★★★★★
- newLISP ★★★★★★
- Perl 5 ★★★★★★
- Bash ★★★★★★

» Python — язык со множественной парадигмой и самый простой в работе.

Расширения языка

Библиотеки, модули... и как заставить их работать.

Все наши языки не настолько богаты классами, как, например, Java, и нередко требуют сторонних библиотек (или модулей, как их иногда называют). Их полнота и управляемость из своего скрипта сильно разнятся.

Perl продолжает нас удивлять потрясающим выбором, имеющимся в CPAN,

но его подход «всегда есть множество вариантов это сделать» частенько приводит в замешательство. Менее очевиден охват расширений *Bash*, созданных для решения проблем: они, вероятно, не идеально подходят для любой реализации оболочки.

Python предлагает отличную поддержку библиотек; соперничающие варианты

очень тщательно рассматриваются сообществом перед тем, как их включают в язык. В каждом решении явно прослеживается озабоченность «правильностью», однако альтернативы остаются в пределах достижимости. По крайней мере, окончательное внедрение менеджера пакетов *pip* в Python 3.4 обеспечило паритет с Ruby.

RubyGems предоставляет формат распространения *gems* для библиотек и программ Ruby, а также *Bundler*, который управляет всеми *gems* по части зависимостей и правильных версий. Вашей единственной проблемой будет поиск наилучшего маршрута по бескрайним библиотекам Ruby. Причем читать придется внимательно.

NewLISP — не очень крупный язык, но выразительный, причем неплохо справляется и без дополнений. Имеющиеся модули и библиотеки адресованы основным потребностям, таким, как базы данных и соединение с Сетью. Этого достаточно, чтобы newLISP стал удобным языком для администратора, но не по сравнению с остальными четырьмя.



➤ Для этого есть не одна библиотека: CPAN — удобный и полезный ресурс для Perl.

Вердикт

Perl 5 ★★★★★
 Python ★★★★★
 Ruby ★★★★★
 Bash ★★★★★
 newLISP ★★★★★

» Долгождность и популярность CPAN отлично сочетаются с хорошей организацией.

Сетевая безопасность

Тестирование и защита сети — или исправление проблем потом.

Тест на вторжения, а то и экспертиза после атаки в небольших компаниях входят в круг обязанностей администратора. Благодаря готовым инструментам вы сможете включить все необходимое в симпатичный скрипт оболочки и держать его под рукой на случай разных ситуаций; но писать перехватчики пакетов или инструменты экспертизы файловой системы в *Bash* не серьезно.

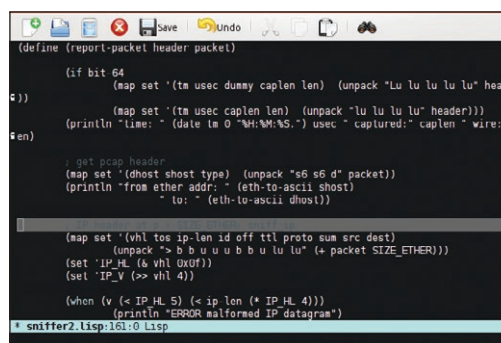
Perl отчасти лишился внимания сообщества после выхода Metasploit, но его инструменты никуда не делись и активно поддерживаются большой группой пользователей, которые не собираются переходить на другой язык. В Perl есть такие инструменты, как *rWeb* — набор инструментов для безопасности web-приложения и тестов на уязвимость — который включен в дистрибутивы типа Kali и Backbox.

Инструменты типа *WireShark* — мощное средство анализа пакетов, но иногда нужно собрать собственный перехватчик пакетов. Python не только предлагает *Scapy*, библиотеку работы с пакетами, но и предоставляет

библиотеку сокетов, позволяя напрямую читать и писать пакеты.

Блоки Ruby (вы пишете функции на ходу, не давая им имен) и прочие функции отлично подходят для написания асинхронного сетевого кода, а его быстрое прототипирование не уступает (и даже превосходит) Python. Однако главный козырь Ruby — Metasploit: это самая популярная программа тестирования на вторжения.

Что касается готовых инструментов, вы можете использовать их вместе и сочетать по мере надобности, однако Perl, Python и Ruby предлагают вам все, чтобы быстро и на ходу проверить сеть на предмет уязвимостей или несанкционированного доступа. Заметьте: Python сейчас фигурирует во всех объявлениях о вакансиях, связанных с системами безопасности.



➤ У newLISP впечатляющие функции работы в сети, хоть ему и не хватает инструментов для проверки на вторжения, которые есть в других языках.

Последний в нашем списке, newLISP, мало известен среди тестировщиков вторжений и серых хакеров, но благодаря встроенной в язык функции работы в сети, вызов функции и несколько аргументов создадут для вас основные пакеты. И снова, у newLISP явно есть потенциал, однако он страдает от относительно небольшой пользовательской базы.

Вердикт

Python ★★★★★
 Ruby ★★★★★
 Perl 5 ★★★★★
 newLISP ★★★★★
 Bash ★★★★★

» Python опережает Ruby и Perl, но все они друзья тестировщиков вторжений.

ЯЗЫКИ СКРИПТОВ

Вердикт

Признаем: очень сложно вырвать вердикт из контекста практики и просто объявить лучший язык. Например, язык *Bash* — отнюдь не самый мощный, и многие части кода, которые значительно экономят время, гораздо лучше объединяются в остальных четырех языках, однако никому из работающих с командной строкой Linux не следует извлекать из изучения скриптов *Bash*.

Perl традиционно рассматривается как следующий шаг, поскольку он тесно связан с командной строкой *nix и по-прежнему повсеместен. Пусть он и проигрывает по сравнению с более новыми языками, но все же не только является всесторонним инструментом Linux CLI, но еще имеет огромное и оказывающее колоссальную поддержку сообщество.

NewLISP — приятный сюрприз. Да, в нем это есть — LISP по части списков — но это компактный язык для встраиваемых приложений и для командной строки. Увы,

его сообщество не отвечает мощи этого языка, и вам следует быть готовым к самостоятельной поддержке своего кода.

Python является мощным языком с поддержкой множества парадигм. Его сообщество — солидное и дружелюбное, и поддерживает все, от образовательных марафонов до обучения преподавателей, а также поддерживает обучение молодежи в Code Clubs и множество других событий.

Однако не менее, чем сообщество, для сисадмина бывают важны наскоро сделанные неряшливые примочки, готовые к скачиванию и многократному использованию примеры, при той выразительности, которая делает многие задачи по программированию если не самыми простыми, то и не самыми сложными. Rails уделяет большее внимание Ruby, но Chef, Puppet и Vagrant позволили напомнить программисту о том, чего можно добиться с помощью

```
private

def vm_host(vm)
  host_options = {
    user: vm['user'] v 'vagrant',
    hostname: vm['hostname'] v 'localhost',
    port: vm['port'] v '22',
    password: vm['password'] v 'vagrant'
  }

  SSHKit::Host.new(host_options)
end
end
end
```

» Мы не устояли перед мощью и очарованием Ruby.

выразительного и функционального языка скриптов, разработанного Юкихио Матсумото [Yukihiro Matsumoto].

Превосходит ли Ruby Python? Стоит ли игнорировать *Bash*? Только не администраторам: они обязаны хорошо знать *Bash*, чтобы следить за происходящим в системе. И кроме *Bash*, каждый сисадмин должен немного знать Perl, Python и Ruby; но нужно еще и глубокое понимание того языка, который вы предпочитаете. **LXF**



«Кроме *Bash*, каждый сисадмин должен немного знать Perl, Python и Ruby.»

I Ruby ★★★★★
 Сайт: www.ruby-lang.org Лицензия: GPLv2 или 2-clause Версия: 2.2.0
 » Мощный, выразительный и очень простой в изучении.

IV newLISP ★★★★★
 Сайт: www.newlisp.org Лицензия: GPL Версия: 10.6.1
 » Очень мощный, заслуживает более частого использования.

II Python ★★★★★
 Сайт: www.python.org Лицензия: PSFL Версия: 3.4.2
 » Изобилие парадигм, упор на правильность и бравое сообщество.

V Bash ★★★★★
 Сайт: www.gnu.org/software/bash Лицензия: GPLv3+ Версия: 4.3.30
 » Делает не всё, и тем не менее остается основным.

III Perl 5 ★★★★★
 Сайт: perl.org Лицензия: GPL or Artistic License Версия: 5.20
 » По-прежнему верный друг сисадмина.

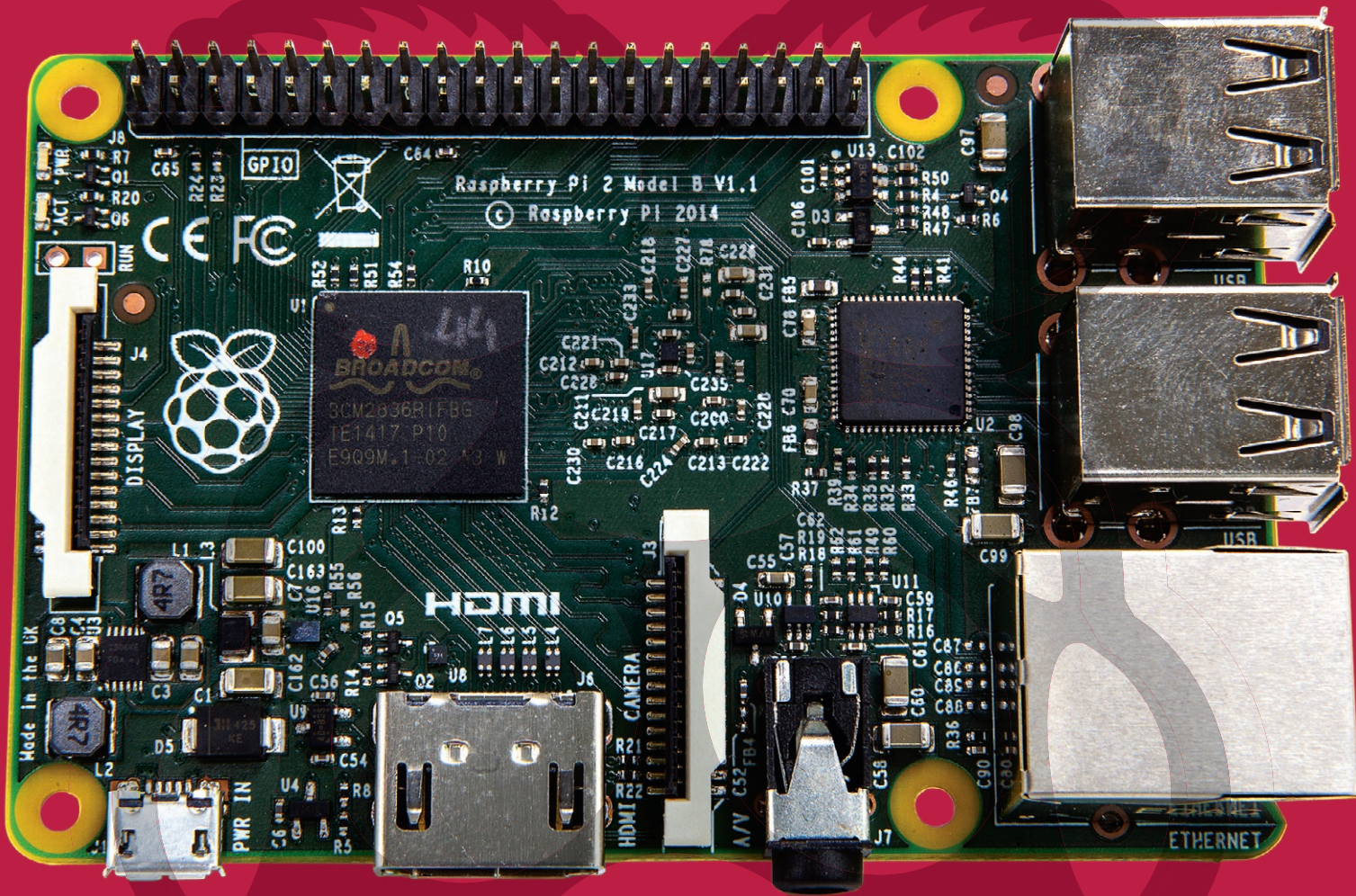
Обратная связь
 Мы не хотим развязывать языковые войны, но хотели бы знать, что применяете вы. Шлите нам свои мнения на lxfl.letters@futurenet.com.

Рассмотрите также...

Хотя в некоторых областях *Bash* отстает, традиционные скрипты оболочки также представлены *Zsh*, который имеет ряд небольших, но полезных отличий: например, лучший доступ к позиционным переменным и способ расширения оболочки посредством функций виджетов. Тем не менее, он не может конкурировать с другими нашими участниками, как и PHP, хотя

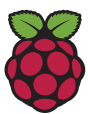
преданные сторонники применяют его в командных скриптах. Наша необычная альтернатива — Rebol (Relative Expression Based Object Language): он перешел в лагерь свободного ПО два года назад, вероятно, поздно для завоевания всеобщей популярности. Однако у Rebol элегантная структура и синтаксис, и удобная консоль цикла read-eval-print (REPL, чтение-оценка-печать).

«Диалекты» Rebol (небольшие, эффективные доменные языки для кода, данных и метаданных) позволяют использовать его практически для всего. Он хорош для обмена и интерпретации информации между распределенными компьютерными системами, но также и для мощных, лаконичных скриптов оболочки. Если вы ищете новый язык для 2015 года, дайте ему шанс.



Знакомимся с Raspberry Pi 2

Лес Паундер дорвался до первой партии Raspberry Pi 2 и ухватил штучку на тестирование.



Новый Raspberry Pi намного мощнее своих предшественников. Это благодаря улучшенному процессору ARM 7, четыре ядра которого работают на 800 МГц каждое, и щедрому 1-ГБ ОЗУ. Это увеличение CPU и ОЗУ — большое преимущество для проектов, сильно нагружающих CPU, как, например, OpenCV и Minecraft.

Raspberry Pi 2 к тому же выигрывает от улучшения конструкции V+, с большим количеством портов USB благодаря LAN9514, который

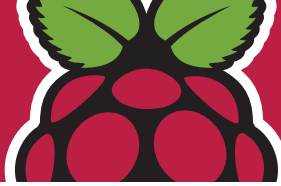
предлагает четыре порта против двух в 9512. Кроме того, V+ оптимизировал управление питанием, и это тоже представлено в Raspberry Pi 2. «Потреб-

«Raspberry Pi 2 выигрывает от улучшения конструкции в V+.»

ление энергии при выполнении определенной задачи сопоставимо с энергопотреблением V+, —

говорит Эбен Аптон [Eben Upton], руководитель Raspberry Pi Trading. — Естественно, чем сильнее нагрузить Pi 2, тем больше энергии он будет потреблять, раз уж делает больше работы. Энергопотребление V+ под большой нагрузкой примерно такое же, как у старой Model B».

Итак, у нас есть самый последний Raspberry Pi — давайте-ка устроим ему тест-драйв! И для нашего расширенного руководства мы возьмем самую новую версию Raspbian, доступную на сайте Raspberry Pi



(www.raspberrypi.org/downloads), поскольку она идет с образом ядра `kernel7.img`, необходимым для ARM7 CPU.

Простейший метод настройки вашей карты microSD — использовать *NOOBS* (New Out Of The Box Software). Для этого нужна как минимум 8-ГБ карта microSD. Скачайте *NOOBS* в виде архива с сайта Raspberry Pi и извлеките содержимое на эту карту (она должна быть отформатирована под файловую систему FAT32). Когда *NOOBS* скопируется на карту, размонтируйте и отсоедините карту от вашего компьютера и установите в Raspberry Pi 2 — вы должны услышать негромкий щелчок, когда она встанет на место.

Подсоедините свой Raspberry Pi к монитору через порт HDMI и затем подключите мышь и клавиатуру через порты USB. Вам также нужно проверить наличие доступа к Интернет для своего Pi. Проще всего это сделать через порт Ethernet. И, наконец, присоедините питание к порту microUSB. Теперь ваш Raspberry Pi 2 загрузится в первый раз.

При первой загрузке *NOOBS* спросит вас, какую ОС установить; в данном случае нужна Raspbian. Выберите ОС и запустите установку, что займет около 10 минут.

По завершении установки Pi перезагрузится и впервые запустит Raspbian, и вы сразу же отметите скорость процесса загрузки, которая теперь занимает всего 17 секунд против 33 секунд для V+! Внимательные также увидят, что теперь при загрузке красуются четыре малинки. Это означает, что у Raspberry Pi 2 четыре ядра — милый сюрприз, возвращающий нас к старым добрым экранам загрузки дистрибутивов Linux.

Сразу после загрузки перед нами появляется меню *raspi-config*, это инструмент для дальнейшей настройки Raspbian Pi. На данном этапе мы просто выйдем из меню и войдем в систему обычным способом. Стандартные подробности входа не изменились, оставшись такими:

```
USERNAME: pi
PASSWORD: raspberry
Войдя в систему, введите
```

```
startx
```

чтобы загрузить рабочий стол. Вы увидите, что он несколько отличается от предыдущих версий Raspbian; это из-за крупных изменений, внесенных Foundation в декабре 2014 г., по большей части дела рук Саймона Лонга [Simon Long], ранее работавшего в Broadcom. Фактически именно Лонг пригласил Эбена Алтона на работу в Broadcom, а сейчас Саймон Лонг работает в Raspberry Pi Foundation над пользовательским интерфейсом, и его первым проектом стало создание нового рабочего стола.

В Raspberry Pi Foundation изготовили исключительно мощный одноплатный компьютер, но как проверить его мощность? Было бы интересно рассчитать число π с удобной нам точностью, скажем, до 10 000-го знака после запятой. Тогда для начала следует установить кое-какое ПО. Откройте терминал и введите следующие две строки:

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberr... x pi@raspberr... x
21238281530911407907386025152274295581807247162591668545133312394804\
04707911915326734302824418604142636395480004480026704962482017928964\
76697583183271314251702909234889627668440323260927524960357996469256\
50493681836090032380929345958897069536534940603402166544375589004563\
28822505452556405644824651518754711962184439658253375438856909411303\
15095261793780029741207665147939425902989695946995565761218656196733\
78623625612521632086286922210327488921865436480229676070576561514463\
20469279068212073883778142335628236089632080682224680122482611771858\
96381409183903673672220888321513755600372798394004152970028783076670\
94447456013455641725437090697939612257142989467154357846878861444581\
23145935719849225284716050492212424701412147805734551050080190869960\
33027634787081081754501193071412233908663938339529425786905076431006\
38351983438934159613185434754649556978103829309716465143840700707360\
41123735998434522516105070270562352660127648483084076118301305279320\
54274628654036036745328651057065874882256981579367897669742205750596\
8340869735020141020672358502007245225632651341055924019027421624843\
91403599895353945909440704691209140938700126456001623742880210927645\
79310657922955249887275846101264836999892256959688159205600101655256\
375676
real    17m25.725s
user    17m25.330s
sys     0m0.020s
pi@raspberrypi ~ $ time echo "scale=10000; 4*a(1)" | bc -l
```

► Расчет числа π до 10 000 знаков — тест для нашего Raspberry Pi. Команда провести тест дается из LXTerminal. Здесь показано время вычисления до разгона.

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

Мы только что установили точный калькулятор, запускаемый из терминала. А теперь проведем тест: рассчитаем π с точностью до 10 000 знаков после запятой, засекая время.

```
time echo "scale=10000; 4*a(1)" | bc -l
```

В нашем тесте на серийном Raspberry Pi 2 расчет потребовал 17 мин. 25.725 с. Мы повторили тот же самый тест на Raspberry Pi V+, и на вычисления ушло гораздо больше времени, 25 мин. 5.989 с. Как вы уже видите, это весьма недвусмысленный знак того, что процессор нового Raspberry Pi 2 намного мощнее, чем в предыдущих моделях.

Наш тест уже продемонстрировал солидные возможности Raspberry Pi 2 в плане производительности по умолчанию; а нельзя ли превратить его в еще более скоростного монстра? Ранее мы закрыли меню *raspi-config*, однако для нашего следующего шага оно нам понадобится. В LXTerminal введите:

```
sudo raspi-config
```

Турбо-наддув вашего Pi

Наша первая настройка после установки — проверка памяти, то есть ее распределения между GPU (Graphical Processing Unit) и основной системой. На Raspberry Pi обычная настройка: около 64 МБ ОЗУ отводится на GPU, а остальное — системе. Это поддается правке, но вы можете все оставить, как есть — или поиграть с этими показателями; в принципе, терминалу не нужно столько памяти, сколько полноценному рабочему столу, и для проектов, ►



Браузер по умолчанию для Raspbian, *Midori*, недавно заменили на *Epiphany*, оптимизированный для Raspberry Pi. Новый браузер доступен через последнее обновление Raspbian и отлично работает на Raspberry Pi 2, да и на более старых Pi.

Ubuntu — на Pi?

Главный сюрприз нового Raspberry Pi 2 — совместимость с Ubuntu для ARM 7 CPU. До выпуска первого Raspberry Pi в начале 2012 г. Ubuntu часто называли кандидатом для Pi, но поскольку Canonical не поддерживала архитектуру ARM 6, которую использовал ARM 11 CPU в исходном Pi, потребовался другой дистрибутив. Сперва для демонстрации возможностей Pi использовался Pidora, ответвление Fedora для Pi. Однако Pidora предлагает полноценный рабочий стол на оборудовании,

требующем более легкого дистрибутива. По мере дальнейшего изучения в качестве достойного кандидата выбрали Debian в форме Raspbian, и он остается официальным дистрибутивом и используется во всех официальных руководствах и документации по поддержке.

Однако совместимость с Ubuntu не означает, что Foundation намерен отказаться от Raspbian: «Мы не планируем официального образа Ubuntu, — говорит Эбен Алтон. — Мы собираемся провести

сравнительные тесты „обычного“ armhf Debian с Raspbian, и, возможно, перейдем на него, если увидим солидный прирост производительности. Мы предпочитаем оставаться с Raspbian — возможно, с динамической заменой основных библиотек, что позволит нам поддерживать Raspberry Pi 2 и Classic из одного образа». На момент написания статьи готовых образов для Ubuntu на Pi не существовало, но в грядущие месяцы их должно появиться немало вам на пробу, включая Ubuntu Core.

где используется исключительно терминал, можно без боязни отвести на GPU всего 6 МБ. Для приложений графического рабочего стола, типа *Minecraft*, потребуется как минимум 64 МБ. Вам предложат перезагрузить свой Raspberry Pi — сделайте это, и вы вернетесь на экран приглашения.

Изменения, внесенные в распределение памяти, позволяют нам вернуться к основному меню *raspi-config* и перейти в меню Overclock. Ваш Raspberry Pi 2 и так работает при 800 МГц на ядро — это уже усовершенствование по сравнению с одноядерным 700-МГц ARM 11 CPU. Мы побеседовали с Эбеном Аптоном и Гордоном Холлингсвортом [Gordon Hollingsworth] о новом CPU, и они оба подтвердили, что его можно разогнать до 1,1 ГГц для каждого ядра. Мы не будем забираться столь высоко, но через меню Overclock разгоним наш Raspberry Pi до стабильных 900 МГц. Это относительно безопасное действие, но не помешает отметить, что если переборщить с разгоном, можно серьезно повредить CPU из-за перегрева вследствие увеличения интенсивности его работы.

Мы спросили команду Raspberry Pi, и те подтвердили, что ядро может нагреваться до 85 градусов, после чего автоматически срабатывает защита и Raspberry Pi выключается. «Экстремалам», желающим разогнать Raspberry Pi 2 до предела, вероятно, стоит приобрести набор радиаторов. Чтобы в любой момент вернуть CPU к обычной скорости, снова войдите в меню *raspi-config* и укажите обычный параметр 800 МГц.

Изменив настройку и несколько раз перезагрузившись, мы успешно разогнали наш новый Raspberry Pi. Давайте запустим

Скорая помощь



Теперь можно смотреть видео с YouTube, благодаря Youtube_dl. Обычно видео на YouTube используют Flash, однако пакетов Flash для Raspbian нет. При просмотре видео на YouTube в браузере, Youtube_dl заменяет Flash элемент web-страницы на совместимое с HTML5 видео.



► Демо *Hello_Pi* — отличный способ продемонстрировать ваш Raspberry Pi 2. Вы можете поместить на чайник любое видео, совместимое с H.264.

графический интерфейс пользователя. Вновь зайдя в систему, введите в *LXTerminal*

```
startx
```

для перехода на рабочий стол. Теперь посмотрим, как наши реформы ускорили вычисление числа π до 10 000 знаков, повторив тест. Откроем *LXTerminal* и повторим код теста, который был

```
time echo "scale=10000; 4*a(1)" | bc -l
```

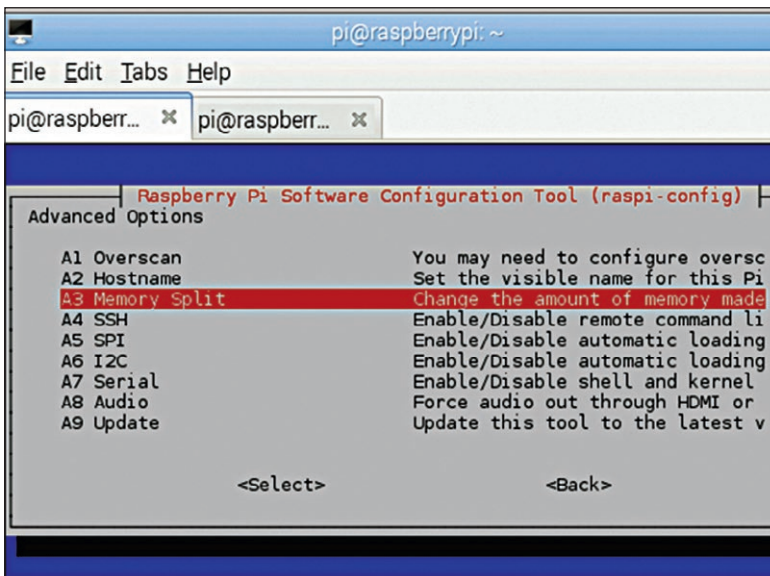
Код запустится, и наш тест теперь займет 15 мин. 28,519 с — ускорение на две минуты!

В Raspberry Pi Foundation позаботились о сохранении преемственности с Raspberry Pi Classic: «Raspberry Pi 2 находился в разработке несколько лет, — говорят Аптон и Холлингсворт, и сюда входит время на разработку BCM2836. — Первый чип появился в начале мая прошлого года; в моем блоге есть видео, где мы с Джеймсом и Домом в лаборатории Broadcom в час ночи, через день после возврата чипа, с демо-видео „на чайнике“, работающим из Linux на плате „Ray“ от Broadcom. Разработка платы Raspberry Pi 2 началась в августе прошлого года (2014), и с октября (2014 г.) у нас пошли образцы. Мы прошли три итерации прототипов, добиваясь нужной производительности».

Совместимость

Производительность отражена в выборе CPU для Raspberry Pi 2. Вместо выбора другой архитектуры Foundation придерживается CPU на ARM, совместимого с ARM11, на котором работал предыдущий Raspberry Pi. Четырехъядерный ARM7 умеет работать с программами, написанными для более старого Raspberry Pi: «Raspbian работает по умолчанию, но требует наличия нового ядра v7, которое будет включено в пакет для скачивания с нашего сайта», говорит Эбен.

Что касается совместимости оборудования, Raspberry Pi 2 использует те же GPIO, что и B+; а значит, платы, произведенные для A+ и B+, поладят и с Raspberry Pi 2. Сюда даже входят платы



► Расширенное меню *raspi-config* содержит опции настройки, позволяющие сделать свой Pi действительно своим, индивидуальным.

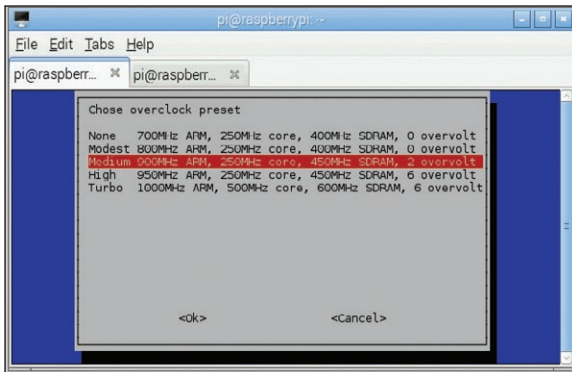
Создание Raspberry Pi 2

В Raspberry Pi Foundation очень довольны новым Raspberry Pi 2. Мы поговорили о разработке с командой инженеров и с Гордоном Холлингсвортом: «Raspberry Pi 2 прекрасен на 100%. Он максимально приближен к обычному ПК, именно этого мы и хотели, когда приступали к работе над проектом». Силы, брошенные на разработку, впечатляют: «В создание нового Raspberry Pi и его процессора команда вложила работу, равную 20-летней, и стоимость такой работы — порядка

£2–3 миллионов». Однако энтузиазм по поводу старого Raspberry Pi не иссяк, как говорит Эбен Аптон: «Есть немало промышленных клиентов, которые не хотят перехода, и, конечно, у нас остается еще Model A+. Чтобы вы получили представление о „привязанности“ к старой платформе, скажу, что после запуска Модели B+ мы продали около 80 000 Моделей B+».

У Foundation есть также и Compute Module [Вычислительный модуль], созданный для встраивания

Raspberry Pi в промышленные приложения. Мы спросили Эбена, будет ли у Compute такое же обновление: «В какой-то момент мы сделаем Compute Module 2, но, наверное, не в первой половине 2015». А как насчет A+? Будет ли обновление у нее? «В настоящий момент ничего не планируется, поскольку цена на A+ — весьма проблемный момент». Итак, пока без обновлений; однако с 2012 г. семейство Raspberry Pi значительно пополнилось, и ныне насчитывает уже шесть устройств.



» В меню *Overclock rasp-config* есть должное предупреждение: переусердствовав с разгоном, вы можете спалить свой Pi.

HAT (Hardware Attached on Top), с пристроенным на плате чипом, который общается с Raspberry Pi с целью быстрой настройки самой же платы.

Однако есть платы, несовместимые с B+ и Raspberry Pi 2 из-за своего размера и конструкции. Такие платы, как PiFace [см. LXF180] и PiFace Control and Display — которые управляли стойкой камеры на Рождественских лекциях Королевского института [Royal Institution Christmas Lectures] — присоединить нельзя. Впрочем, команда OpenLX SP, занимающаяся этими платами, не поленилась выпустить их специальные версии для B+ и Raspberry Pi 2.

Тест 3D-графики

Все Raspberry Pi поставляются с одним и тем же VideoCore IV GPU (Graphical Processing Unit), который позволяет Raspberry Pi воспроизводить видео с высоким разрешением 1080. Новый Pi, кроме того, идет с GPU, тоже созданным Broadcom, как и BCM2836, на котором новый Pi работает. А вы знали, что существует тестовый пакет для GPU?

Вы можете найти тестовый пакет, открыв *LXTerminal* и введя `cd /opt/vc/src/hello_pi/`

Обнаружится ряд директорий, содержащих множество разных демо-видео. Но для их использования надо скомпилировать демо из исходника. С целью упрощения этого процесса Foundation предоставила автоматический скрипт компиляции, запускаемый только однажды. Чтобы запустить скрипт, введите в *LXTerminal* команду `./rebuild.sh`

Запустится процесс компиляции для всех демо-версий, так что это может занять несколько минут, даже на нашем новом разогнанном Raspberry Pi.

По завершении вы получите ряд готовых к запуску демо-версий, и первой в списке должна быть *hello_teapot*. Чтобы запустить ее, в *LXTerminal* убедитесь, что вы все еще в директории *hello_pi*, и введите

```
cd hello_teapot
./hello_teapot.bin
```

Теперь вы увидите 3D-модель чайника с видео, привязанным к его поверхности. Чтобы выйти из демо чайника, удерживайте **Control+C**, и вы вернетесь в *LXTerminal*.

Еще одно демо-видео, которое стоит попробовать — *hello_triangle2*, и чтобы его найти, вернитесь в директорию *hello_pi* — это делается вводом

```
cd ..
```

Из *hello_pi* мы можем сменить нашу директорию на *hello_triangle2* и запустить демо, введя

```
cd hello_triangle2
./hello_triangle2
```

На первый взгляд это демо покажется довольно-таки статичным, но поведите по нему мышью, и вы увидите два наложившихся друг на друга фрактала, которые движутся и реагируют на перемещения мыши. Похоже, фракталами можно управлять, чтобы создать идеальную окружность. Для выхода из демо *hello_triangle2* удерживайте **Control+C**, и вы вернетесь в *LXTerminal*.

Итак, мы рассмотрели новый Raspberry Pi 2, увидели, как он работает по умолчанию, и постарались дополнительно разогнать наш Pi. На следующей странице мы попробуем поработать с *Minecraft* через PiBrella, чтобы создать кнопку, по которой срабатывает система бомбометания!



У Raspberry Pi 2 те же размеры, что и у B+, но для тех, кто хочет снова использовать корпус B+, например, для PiBrow, стоит отметить, что некоторые компоненты монтирования интерфейса перемещены. Эти изменения не влияют на общий размер платы, но поскольку PiBrow использует уровни компиляции, для вашего PiBrow потребуются новый уровень.



Обновляем свой Pi

Raspberry Pi Foundation выпустила множество обновлений к существующему Raspbian, и постоянное обновление вашего Raspberry Pi — это очень хорошая практика.

Есть несколько удобных команд Terminal, которые вам помогут — например,

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

проверит наличие самого последнего ПО, а

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

похожа на **upgrade**, но это более умная программа обновления: она удаляет старое ядро.

Если вы бы хотели установить новый рабочий стол — а кто бы не хотел, ведь Саймон Лонг отлично над ним поработал! — введите в терминале следующие три строки:

```
sudo apt-get update
```

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

```
sudo apt-get install raspberrypi-ui-mods
```

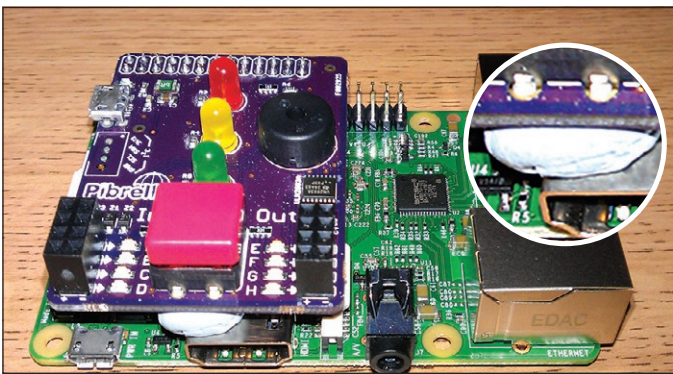
По завершении, перезагрузите свой Raspberry Pi, чтобы изменения вступили в силу. Когда вы снова войдете в систему, вас будет приветствовать новый улучшенный интерфейс.

Обновление делается через терминал с помощью мощнейшей утилиты управления пакетами *apt* — ее используют все дистрибутивы на базе Debian.

Объединяем Pibrella с Minecraft

1 Подключим Pibrella к Raspberry Pi

Мы будем использовать большую красную кнопку в Pibrella.com, чтобы взрывать TNT в *Minecraft*. Pibrella присоединяется через первые 26 контактов GPIO. Не присоединяйте плату при включенном устройстве! Используйте пластилин Blu Task или нечто подобное, чтобы не допустить замыкания нижней части платы и HDMI. Теперь присоедините остальные кабели, как обычно, но питание подведите к USB-порту.

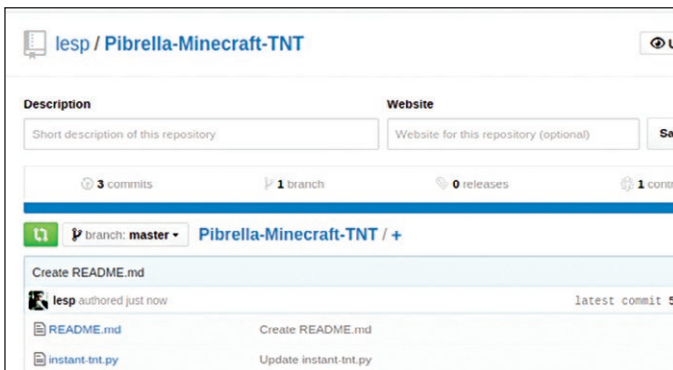


3 Получим код

Мы создали репозиторий GitHub, содержащий код для нашего урока; загляните на <https://github.com/lesp/Pibrella-Minecraft-TNT> и скачайте себе копию. Затем откройте *LXTerminal* и введите

```
sudo idle
```

Откроется *idle*, редактор Python, с повышенными привилегиями, позволяющий нам использовать Pibrella из Python. Теперь откройте пример кода.



5 На позициях

Minecraft использует координаты *x,y,z* для уточнения положения объектов в мире. Мы создаем функцию под названием `button_changed()`, которая определяет положение игрока и затем создает куб TNT в координатах рядом с игроком. И, наконец, настраиваем функцию, вызываемую при нажатии на кнопку. Оставьте окно открытым, запустите *Minecraft* и создайте новый мир.

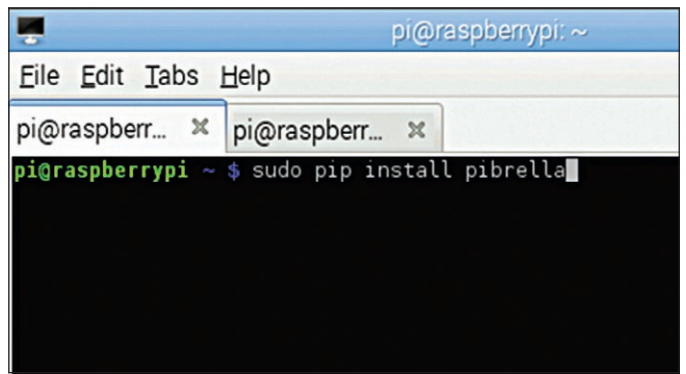


2 Настроим Pibrella

Загрузив рабочий стол на наш Pi, откроем *LXTerminal* и введем:

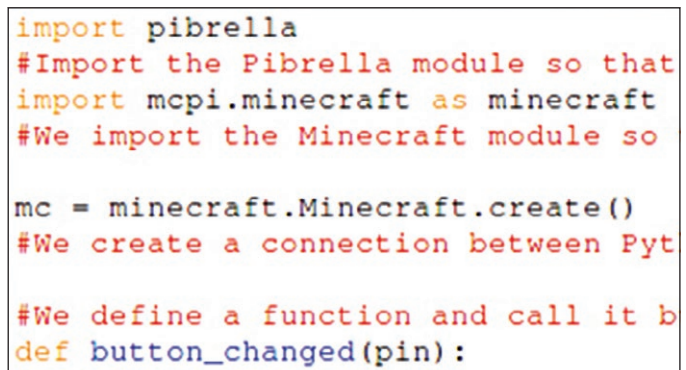
```
sudo apt-get update
sudo apt-get install python-pip
sudo pip install pibrella
```

Установятся программы, нужные нам для использования Pibrella из Python.



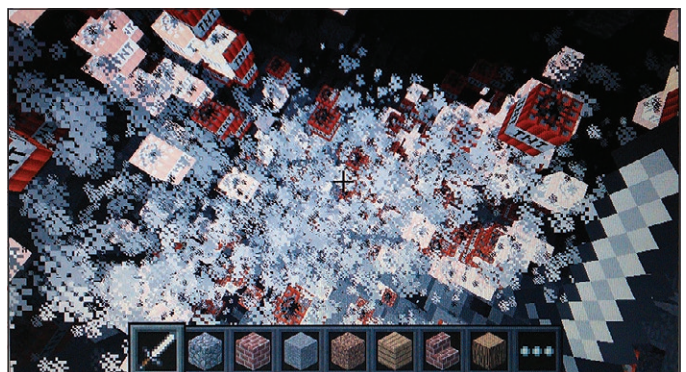
4 Изучим код

Наш код написан на Python 2, поскольку модуль *Minecraft* пока доступен только для этой версии. Читать код достаточно просто. Помните, что строки, которые начинаются с # — это комментарии. Первые несколько строк — импорт в наш проект дополнительных функций в виде библиотек Pibrella и *Minecraft*. Затем мы воспользуемся переменной с именем `mc` для хранения информации о соединении с *Minecraft*.



6 Сбросим бомбу

При *Minecraft* в готовности и открытом нашем коде, нажмите на TAB, чтобы освободить мышь из *Minecraft*, и щелкните по Run > Run Module in *idle*. Появившаяся оболочка *idle* запустит код. Переключитесь снова в *Minecraft* и перейдите в уютное место. Нажмите на красную кнопку Pibrella, чтобы сбросить бомбу. Поразите TNT своим мечом... и БЕГИТЕ! Помните: можно запустить это и на обычном Pi, но не исключен отказ *Minecraft*. LXF



26-27 мая 2015 года

«Ярмарка вакансий онлайн» для студентов и молодых специалистов

Москва,
Санкт-Петербург,
Екатеринбург,
Новосибирск,
Томск



www.expo.hh.ru

Уинн Незерленд

Сделал Уинн

Мэтью Хэнсон беседует с Уинном Незерлендом о важности API и тестирования и прелестях Ruby.



Уинн Незерленд (Wynn Netherland) помог сделать Всемирную сеть такой, какой мы привыкли её видеть последние 20 лет. Плодотворный разработчик и майн-

тейнер оболочек API Ruby, ныне он работает над API GitHub, а также является автором целого ряда книг.

Linux Format: Какие изменения в Интернет за те годы, что вы им занимаетесь, вы считаете крупнейшими?

Уинн Незерленд: Отличный вопрос. Пожалуй, он стал во многих отношениях профессиональнее: нет уже той ковбойской бесшабашности. В былые дни, когда я работал в Compaq, а затем в Hewlett-Packard, мы были своего рода пионерами, продвигавшими Open Source в сфере электронной торговли, по мере сил и возможностей. Тогда все ещё только пытались понять, как это — совершать платежи по Интернету. Не было RoboSource Control и всего прочего, что сегодня кажется само собой разумеющимся. Да и взять даже тогдашние законы. Так что Интернет с тех пор очень вырос, стал намного... организованнее. Во многом мы потеряли ту... хотел сказать — невинность, ту атмосферу игры и юношеского задора, когда все только начали понимать, что к чему, и не было таких жёстких рамок, за исключением AOL! Впечатление создаётся такое, что мы со временем во многом жертвовали открытостью Всемирной сети, увеличивая число «запертых садов». И по непонятной причине нас это в основном устраивает. Уж и не знаю, почему.

LXF: Наверняка раньше молодые люди начинали с того, что заводили себе учётные записи Geocities, где можно было создать собственный сайт, поэкспериментировать с HTML и даже внедрить систему оплаты, если вы что-либо продаёте. Стартовать было легко. А теперь люди, вместо того, чтобы учиться создавать сайты, думают только о том, как завести себе блог Wordpress или страницу на Facebook.

УН: Да, и похоже, что второй вариант популярнее первого. Ведь выбирая тему на Wordpress или используя другой аналогичный сервис, или открытые инструменты CMS, вы рано или поздно сообразите, что имеете дело с HTML или CSS. А с Facebook что видишь, то и получаешь.

LXF: Значит, по вашим словам, в Интернете стало больше ограничений, но всё же что-то изменилось в лучшую сторону?

УН: Как и во всём, что делает человек, есть и плюсы, и минусы. С той же совместимостью: с одной стороны, возросла сегментированность данных, с другой — улучшились возможности обмена. В конце 1990-х мы до сих пор работали с такими вещами, как Corba, а XML ещё толком и не было. Приходилось посылать HTML в браузер и уповать на то, что всё отобразится правильно! А сейчас у нас есть инструменты, донельзя упростившие обмен данными; JSON и HTTP лежат в основе



всего Интернета. Но кому принадлежат эти данные? Как мне их экспортировать и забрать себе? Это уже проблемы из разряда более серьёзных. Наиболее актуальная из них, пожалуй — кто может видеть мои данные.

LXF: Да, это насущно. Учитывая ваш личный интерес к пользовательской стороне API, которую это также затрагивает, почему это так важно?

УН: Моё изучение опыта пользователей началось с работы в газетной типографии, впоследствии я стал более углублённо заниматься этим вопросом. И когда несколько лет назад появились открытые API, мне захотелось самому попытаться

«как бы я сам хотел это видеть», или «как сделать так, чтобы с этим инструментом было удобнее работать».

LXF: Очевидно, что для API очень важна стабильность — насколько важна, по-вашему?

УН: Это крайне важно. Я думаю, это в чём-то идеалистическая цель. Пользователю легко сказать: «Эй, вы сломали мне приложение! API менять нельзя!», верно? У нас в сообществе API есть голосистовое меньшинство, позиция которого — если у API возможны версии, значит, он неверен в корне. Штуки вроде HyperMedia и прочих медиа помогают отделить изменения, но всего ведь не пре-

О ТОМ, КАК ИЗМЕНИЛСЯ ИНТЕРНЕТ

«Мы со временем во многом жертвовали открытостью Всемирной сети, увеличивая число „запертых садов“.»

выудить данные из тех систем, которые я использовал. Собрать в свой блог всё то, с чем я работал в Интернете. Меня просто возмутило, что пользователям чуть ли не умышленно не дают выкачать их же данные! Что объясняется обычным несочувствием их интересам. В сущности, удобство пользователя и сводится к сочувствию. Мы привыкли считать интерфейс пользователя непременно графическим. Практически сводя GUI к 'G'. Тогда как Unix работает именно через текст. И это тоже интерфейс пользователя. Только, по большей части, без графического элемента. Так же и с API: даже если настроить REST (считать состояние клиента), невозможно представить, как будет выглядеть API с клиентской стороны, чтобы не просто погрузить в REST свою базу данных, а задуматься

дусмотришь, я не думаю, что изменений можно избежать всегда... Иногда они возникают в результате исправления ошибок или приведения в соответствие с уже работающим кодом. Допустим, вы не заметили ошибку, опубликовали API, и он кое-как работает. И целые группы людей успевают создать собственные инструменты поверх этой ошибки. Если вы её, в итоге, обнаружите и исправите, эти инструменты поломаются.

LXF: Значит, важно сочувствовать пользователям API, но есть некий предел, когда нужно делать по-своему?

УН: Совершенно верно. Поэтому в GitHub API мы начинаем с подсчёта изменений. Тех изменений, которые нам предстоит внести. Будь это »

рефакторинг, добавление новой оконечной точки или удаление таковой, или тому подобное. Мы всё это измеряем и подсчитываем. А затем ищем способ минимизировать изменения. И таким образом решаем эту проблему. Так, если вас есть оконечная точка, которую никто не вызывает, значит, её можно безболезненно удалить. На самом деле, здесь всё завязано на человеческом факторе. Решение убрать какую-нибудь оконечную точку может создать проблему для команды Ops и для инфраструктуры, а порой и угрожать работе сайта в целом — даже до такой степени. Мы стараемся минимизировать этот риск, и если дело доходит до исключения какого-то метода API, мы прежде всего выясняем, кто и как часто его использует, и обращаемся к ним, говоря: «давайте мы вместе подумаем, каким ещё способом можно решить эту задачу».

LXF: Вы упомянули GitHub. Как вы туда попали?

УН: Когда работал над оболочкой библиотек API. Несколько лет назад я был сотрудником подкаста Changelog, и хотелось, среди прочего, продемонстрировать, как много делается на GitHub в формате Open Source. Мы называли это Tale of the Changelog [Сказ о Changelog]: по сути, это был обзор всего происходящего на GitHub. Из тогдашних оболочек Ruby мне не нравилась ни одна — они слишком близки к REST, а я убеждён, что оболочки API должны быть идиоматичны языку, на котором написаны. Вот если это Ruby, она должна ощущаться как расширение Ruby. Поэтому я создал то, что потом превратилось в Octokit, а когда через два года меня позвали в GitHub, вдруг выяснилось, что там она вовсю используется. Знай я это раньше — вложил бы в неё больше усилий! Ведь я-то делал этот проект только ради интереса. Когда я впервые загрузил исходный код GitHub и он стал устанавливать Octokit, я был

в шоке! Так что теперь это основной интерфейс API для многих наших инструментов, поскольку они написаны на Ruby. Он отвечает за проверку разрешений внутренних скриптов, получение статистики, аутентификацию различных внутренних приложений. А в основе всего этого лежит Octokit.

LXF: Должно быть, лестно было увидеть там свой код...

УН: Было. Но любой открытый проект — это плод коллективных усилий, так что за эти годы много кто приложил к нему руку. Для Octokit больше всех сделал, пожалуй, Эрик Михаэлс-Обер [Erik Michaels-Ober]. Он работает в OpenCloud и создал для них ряд оболочек. Но затем команда Mac захотела для своего приложения Mac изъять из GitHub оболочку API для Objective-C, и им было нужно как-то её назвать. Создание имён, как и аннумирование элементов кэша, в вычислительном мире составляет большую проблему, верно? Было очевидно, что лучшее название для клиента Objective-C — это Octokit, поэтому я сказал им: «Ребята, это имя должно быть вашим, нам нужно просто сделать из него общий бренд, раз лучшего варианта нет»; так наш проект стал называться Octokit, а я поменял своё пространство имён в GitHub — на Octokit Org, так что теперь у нас есть вариант и на Objective-C, и на .net. Меня постоянно спрашивают, почему нет на Python. На самом деле, есть вариант на Go, но главный принцип создания оболочек Octokit в том, чтобы они были первой инстанцией — мы их используем, тестируем и поддерживаем. И потому не занимаемся чем-то, что постепенно устаревает и перестаёт использоваться.

LXF: В чём состоят ваши главные обязанности как члена команды API в GitHub?

УН: Главным образом, в поддержании её на плаву. Просто на данный момент в команде API нас только двое, Джейсон Рудольф [Jason Rudolf] и я. И она входит в состав другой, большей команды, именуемой '.com engineering team'. Мы отвечаем за горизонтальные аспекты API, по всему сайту. Такие вещи, как аутентификация, ограничение скорости и всё прочее, что обязан применять разработчик API. Кроме того, мы обучаем и поддерживаем разработчиков функций в решении проблем с запросами на включение изменений и доступом к репозиториям для расширения и сопровождения их API, в развитие того, что они делают на сайте. API — это маленькое приложение внутри большого web-приложения, поэтому мы публикуем много внутреннего кода, изолируя только способы его обработки и вывод.

LXF: Значит, вы плотно занимаетесь и тестированием поведения API, и в своих прошлых интервью упоминали некоторые инструменты, которые вы для этого используете, например, JSON. А вы можете пояснить, что представляет собой процесс тестирования?

УН: Конечно, основной формат вывода данных API — JavaScript Object Notation (JSON), поэтому мы используем схему JSON. Одним из недостатков JSON, в отличие от XML, является отсутствие какой-либо заданной структуры. Здесь легко разработать синтаксис, спецификации самые примитивные и доступные на том языке, который вы выберете, но неизвестно, как будут выглядеть данные. Поэтому мы используем схему JSON как для ввода, так и для вывода. Со вводом мы работаем совсем недавно, и уже добились больших успехов. Традиционно, в случае с выводом, в ходе проверки мы можем по виду возвращаемого JSON обозначить свойства отклика объекта и тому подобное. Теперь мы можем делать то же и при импорте. Создавая запрос на включение, вы передаёте нам некий JSON, мы пропускаем его через систему проверки и говорим: «Извините, это неправильный формат». И благодаря этому мы можем декларировать в тесте, как выглядит правильная форма документа. Это избавляет нас от необходимости писать на каждую мыслимую перетасовку отдельный тест.

LXF: Вы также используете Rack::Test...

УН: Rack::Test применяется для всех приложений Rack, а Rack в мире Ruby — это HTTP-конвейер. Несколько лет назад многие платформы объединились ради создания стандартизированного конвейера ввода-вывода, получившего имя Rack. Теперь же многие используют этот интерфейс, позволяющий надстраивать друг на друга — как в стойке серверов центра обработки данных — межплатформенные приложения и работать с ними в формате конвейера. То есть если речь идёт об операциях, где используется HTTP-отклик, он обрабатывается, а затем передаётся другим приложениям. Например, при аутентификации, ограничении скорости, действиями с cookie, когда вам нужно проверить ввод и вывод на межсайтовые уязвимости, и всякое такое.



Мы используем Rack::Test для тестирования своих приложений.

LXF: Важно выявить потенциальные проблемы уже на этой стадии, верно? Как же вам удаётся отследить способы использования ваших API? Возвращаясь к разговору о сочувствии нуждам пользователя, это действительно очень важно.

УН: У нас есть кое-какие метрики на разных уровнях. Во-первых, это сами отклики и их статус в определённых оконечных точках. У нас есть оконечная точка События [Events] — это наш источник активности API. Там накапливается некий объем информации, а мы знаем, что это от приложений, создаваемых на базе GitHub. То есть на первом уровне показателем является сам факт такого объема, но он не всегда даёт полную картину. Помимо него — видимо, потому, что не все методы одинаково ценны — меня больше всего привлекают более углублённые задачи, вроде создания запросов на включение, их объединение и другие действия, обычно выполняемые на сайте. Не просто использовать API для хранения окрошки из данных в базе данных — мне нравится видеть, как приложения выстраивают на них целый рабочий процесс: копируются репозитории, добавляются функции, а затем через API создаются запросы на включение. Вы увидите массу инструментов для создания приложений пересмот-

результаты и что-то зафиксировать в диаграммах и графиках. Сопоставив их, вы поймёте: вот это — основная ветка, а вот это — новая. В 10% времени они отличаются, и нам надо понять, откуда идут эти расхождения. Или же новая ветка совпадает на 100%, но выполняется медленнее или быстрее. Science позволяет сравнивать такие вещи, чтобы добавлять этот код с уверенностью, основываясь на конкретных, проверенных данных, а не тыкать пальцем в небо, надеясь, что «авось, сработает!»

LXF: Надо думать, что это также помогает вам найти типовые решения?

УН: Да, особенно когда речь идёт о безопасности, где просто нет права на ошибку.

LXF: И давно вы используете Science?

УН: Science уже... в том виде, в каком он есть [смеется]... наверное, примерно год. И он не всегда был открытым. Впервые я заговорил об этом в марте 2014 года, и в команде яро подхватили эту идею, так как код устаревает, и мы начинаем искать способы сделать его побыстрее и поновее.

LXF: Насколько важны инструменты Open Source для создания и тестирования API?

УН: По-моему, без них никак не обойтись. Другое решение вне экосистемы Microsoft просто нет.



те же самые приложения, просто обрабатываемые в разных системных форматах, верно? Ну вот, число приложений, подключённых к Сети, значительно возросло, теперь это не только прерогатива браузеров. Что также изменило наше представление о родных приложениях. Помню, как я создавал свою вторую или третью оболочку для API для ныне не существующей [местной соцсети] Gowalla. Тогда это был конкурент Foresquare. Если в Foresquare нужно было соревноваться за статус мэра какого-нибудь места, в Gowalla было нечто вроде сбора наклеек на чемодан. То есть за посещение любого штата США или любой страны мира вы получали особый значок. Ещё можно было оставлять виртуальные объекты для других. Там была совсем другая атмосфера, всё было настолько артистично, мне очень нравилось. У меня там работали друзья, я их всё время подначивал сделать API, а они всё время говорили, что работают над этим, но как-то таились. Потом у них вышло мобильное приложение, и я создал оболочку Gowalla API, разувзав их мобильный API и запустив их приложение через прокси-сервер. Я также накопил там много разумных решений, которые потом стали типовыми в других современных API, в том числе и в GitHub.

LXF: А когда вы пытаетесь представить, как разные люди будут работать с определёнными сервисами или приложениями, это же, наверное, тоже сложно протестировать?

УН: Есть у нас один способ, раз уж изменения не дешёвы и зачастую, как мы уже говорили, нежелательны! Мы ввели идею бета-версий оконечных точек для API, обычно это выглядело так: API выпускают, немного погодя становится понятно, что в нём нужно поменять, и появляется вторая версия API — в GitHub так и делалось, когда был V2 »

О ПРОТИВОПОСТАВЛЕНИИ СЕТЕВОГО И ИСХОДНОГО

«Это ложная дихотомия, поскольку локальные приложения тоже используют JSON и API.»

ра кода на базе GitHub, которые делают примерно то же, что мы делаем с сайтом, но с учётом особенностей решаемых задач.

LXF: Ещё одним открытым проектом, о котором вы упоминали, был Science. Можете рассказать нам о нём?

УН: Это детище Джона Барнетта [John Barnette] и Рика Брэдли [Rick Bradley], оба из GitHub. Была у нас одна проблема в кодовой базе .com, что при наличии прекрасного тестового комплекта, уникального, по сути, основанного на опыте других моих проектов, были ещё некоторые аспекты самих приложений, довольно уязвимые, в которых нельзя было быть уверенным на все 100%. Например, когда применяются сложные техники метапрограммирования на динамическом языке, требующие отслеживания на этапе проектирования. То есть, когда есть нечто критическое, что нельзя изменить, но предполагается реорганизация кода, требуется Science — это приложение... точнее, проект... который даёт подсказку тестировать альтернативные ветки кода. Если используется метод только на чтение, ничего не меняющий, и приложение возвращает известное вам значение, как и при научном методе, вы можете запустить обе ветки кода параллельно, сравнить

Первый же слайд к моему докладу [на прошлогоднем OSCON] был посвящён созданию отказоустойчивых API в среде Open Source, и тут же завязался долгий разговор... Сейчас у меня всего 40 минут на разговор, так что я вынужден многое опустить! Но на этом действительно завязано всё — web-архитектура, серверы, на которых будут использоваться API, средства сбора метрик и обратной связи о работе сайта. Методы тестирования... каждый из этих аспектов — целая матрица решений, выборов, какие инструменты использовать. А в сообществе Open Source такой выбор, что просто глаза разбегаются. Тирания выбора, и приятно иногда прийти в проект, как было у меня с GitHub, где многое уже заведомо определено и можно просто играть по этим правилам.

LXF: Возвращаясь к API, сейчас есть целый ряд устройств для доступа к Интернету и онлайн-ресурсам. Создает ли это какие-либо трудности в разработке API?

УН: Безусловно, растёт нагрузка на API. Недавно узнал, что некоторые люди, говоря о мобильных приложениях, разделяют «сетевое» и «локальное», но это практически ложная дихотомия, поскольку локальные приложения тоже скрыто используют JSON и API. По большей части, это

и совершенно другое пространство имён URLs, а потом вышел V3. С переходом на V3 мы решили, что версии окончательных точек нужно привязывать не к пути, а к выводу. То есть вы отправляете некий заголовок HTTP и тем самым запрашиваете конкретную версию ресурса. Испробовав этот метод в течение 18 месяцев, мы стали использовать его как бета-пароли для новых функций. Выпуская новую версию API, мы прямо в заголовке прописываем слово 'preview', и это ваше условное соглашение: «Я понимаю, что данная окончательная точка API не годится для разработки поверх нее», но так мы можем получить обратную связь от реальных пользователей. У этих бета-API нет гарантийного периода, мы в любой момент можем что-то поменять, но мы уже делали так пару раз, и это оказалось очень полезно, поскольку народ стал использовать некоторые функции совсем иначе, чем мы предполагали. И потому внесли изменения ещё до выпуска готового продукта, тем самым повысив его качество.

LXF: Значит, начало было обещающим?

УН: Да. И не только в отношении бета: теперь, если выясняется, что какой-то ресурс неудобен для целого класса пользователей, мы можем предложить им те же данные в другом формате, который им подойдёт. Например, когда из приложения удаляются определённые функции, бывает необходимо удалить некоторые поля, а в Python или Ruby удалить поле в JSON и даже установить его в ноль — проще простого. В статических клиентах это менее удобно. Из-за проблем с компиляцией. Поэтому в таких случаях мы можем предложить другой вариант той же оконечной точки, позволяя им выбрать ту версию, которая их устроит.



LXF: Вы сказали, что в GitHub, в основном, предпочитают Ruby?

УН: Да, пожалуй, большая часть нашего инструментария — на Ruby. Всё больше используем Go, для загрузки медиа и тому подобного, просто потому, что он хорошо подходит для параллельной передачи. Приложения для Mac делаются на Objective-C, а теперь на Swift, специалисты сейчас его оценивают и постепенно переходят на него. А команда Windows, само собой, работает на .net. Так что мы, как всегда, полиглоты, но для web-приложений в приоритете Ruby.

LXF: Чем же так удобен Ruby?

УН: Ну, как бы ключом к сообществу Ruby является «Путь к счастью — в оптимизации [optimizing for happiness]». Просто Ruby... вот я пришёл в это сообщество из .net, и лично для меня ничего лучше, чем Ruby, и быть не может. Иногда необходи-

тогда только что перешли на Subversion с применением *Git*. И весь стол вдруг подхватил: «Да, тебе стоит попробовать, знаешь, как мне это помогло», и когда видишь такие моменты озарения, и то, как целые проекты переходят на GitHub, то становишься просто частью этой приливной волны, которая вливается туда и затем выплескивается обратно. Очень важно, что в большинство проектов вход беспрепятственный. Репозиторий Canonical — один из самых активных, и если ваш проект впал в спячку, ничто не мешает сообществу ответить его — и он уже часть Canonical, где жизнь кипит. Те, кто занимаются поддержкой, должны прилагать усилия, чтобы удержаться на вершине своих сообществ. Или передать это право другим. Потому что мы не вечны, и раз уж подобная деятельность возникла, вскоре появится новая модель — кто-то пришлёт запрос на включение, и, видя, что он хороший, добротный, вы делаете то, что ещё

О ПРИЛОЖЕНИЯХ, КОТОРЫЕ ЗАПОЛНЯТ МИР

«Вскоре мы окончательно потеряем счёт программам, они будут внедряться в наше окружение.»

ма большая производительность, на какую Ruby не способен, поскольку это динамический язык, хотя его и можно разогнать. Мне очень по душе Go, но по простоте ему с Ruby не сравниться. И это довольно интересно, ведь я много писал и на JavaScript, но синтаксис Java, со словом Function на каждом шагу, дальше от человеческой речи, чем Ruby... Его средства метапрограммирования открывают интересные возможности для работы со специфическими языками... вместо того, чтобы писать кучу утверждений, я могу уместить это в пять строк DSL, которые будут всё это включать. В нём всё просто и понятно без лишних подробностей, которых так много в других языках.

LXF: И он гораздо дружелюбнее к новичкам.

Не нужно ждать до самого конца, чтобы увидеть результат...

УН: Он очень легко даёт, особенно новичкам... из недостатков только то, что в Ruby некоторые вещи можно порой сделать тремя разными способами, и вы можете просто не знать, какой из них самый общепринятый. Так что научиться с ним работать может быть сложнее, чем понять его принципы, но он легко даёт.

LXF: Недавно GitHub объявил, что у него уже восемь миллионов пользователей, то есть это крупнейшая кодовая база в мире; что, по-вашему, стало причиной такого успеха?

УН: Open Source и привлечение ключевых разработчиков и ключевых проектов сделали GitHub феноменальным. Помню, как когда-то, ещё в период бета, чтобы заманить кого-то на GitHub, нужно было предварительно заманить его на *Git*. Что и сделал однажды один джентльмен, который как-то оказался рядом со мной на ланче: они

недавно казалось неслыханным: вы передаёте право управления другим, чтобы они стали частью этого проекта, поскольку поддерживать открытый проект в одиночку невозможно — это только затормозит его развитие.

LXF: Думаю, советы открытого сообщества приносят пользу.

УН: Меня до сих пор поражает, что стоит выйти за пределы традиционных технических конференций — в особенности в дизайнерские круги — где применяется всё больше и больше HTML5 и других технологий, и оказывается, что взаимодействие с сообществами разработчиков Open Source уже идёт на уровне мировой общности. Или попасть в корпоративную среду, гораздо более закрытую, но также развивающуюся в данном направлении — очень здорово бывает окунуться в эти сообщества и увидеть, как они внезапно переживают озарение: «Ух ты, это же поможет оживить наш проект».

LXF: Что ждёт GitHub в будущем?

УН: Он перевернёт мир приложений. Думаю, уместно будет сказать, что программное обеспечение заполонит мир. Учитывая, как глубоко оно проникло в нашу жизнь. От устройств, которые мы держим в руках, до жизненно важных, включая больницы. Лично я боюсь не уколов, а приложений! Я-то знаю, как они пишутся! [Смеется.] Но программы уже повсюду, верно? И вскоре мы окончательно потеряем им счёт, они будут всё больше внедряться в наше окружение, и тем больше потребуются таких инструментов для всех сфер индустрии. Полагаю, GitHub будет подходящей взлётной полосой для подобной деятельности. **LXF**

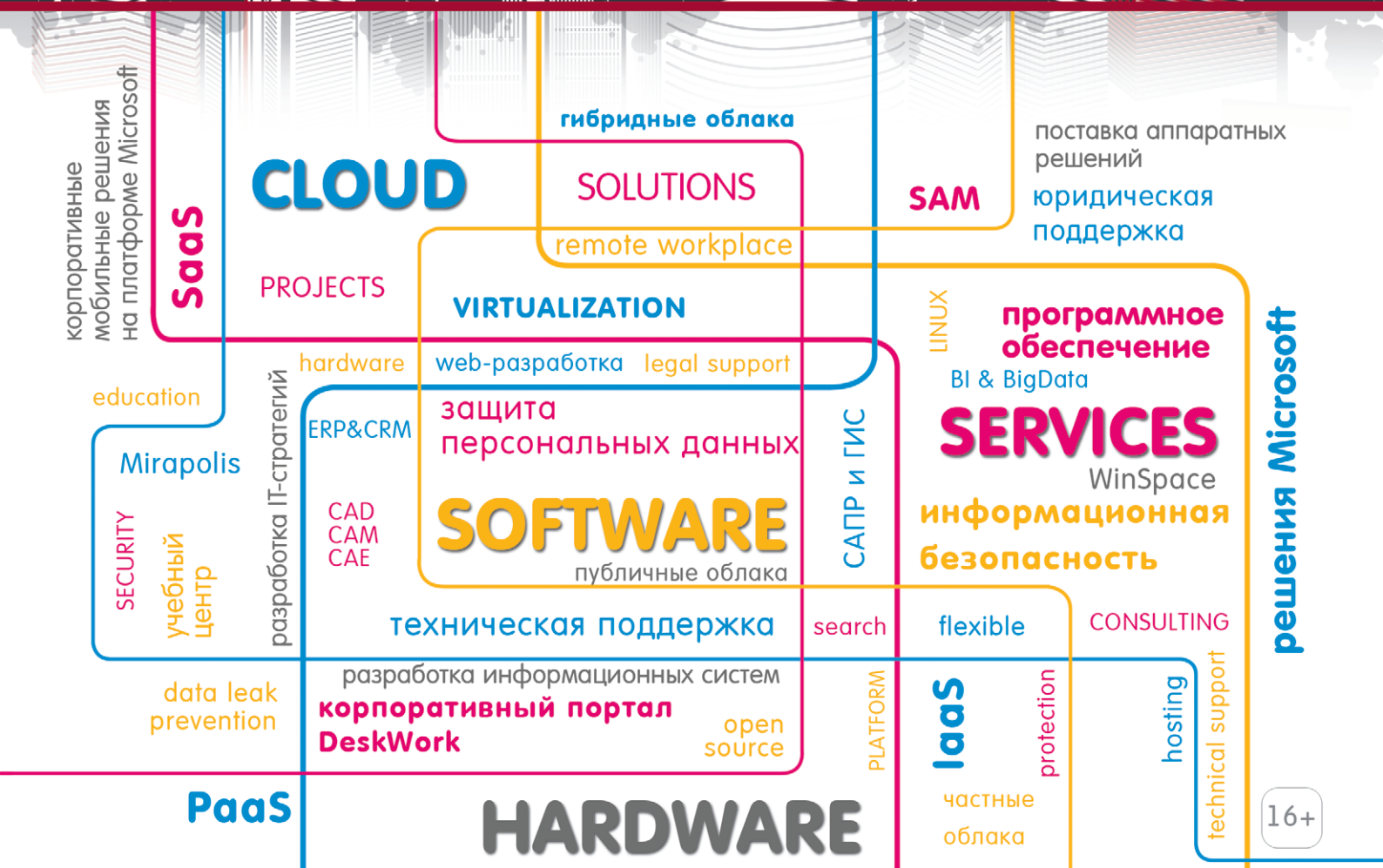
softline®

Cloud Software Hardware Services

20+

Years in IT

IT-архитектура вашего бизнеса





Идем в Linux

Вам любопытно, что такое Linux, но непонятно, как зайти на эту незнакомую территорию? Маянк Шарма поможет вам влиться в растущие ряды линуксоидов.

Сейчас — самое удачное время приступить к освоению Linux. Он удобнее и проще в установке, чем когда-либо, а все наиболее значимые дистрибутивы, типа Ubuntu, недавно обновились. Поэтому Linux Format решил, что пора попробовать начать всё с чистого листа, с новой установки.

Если вы никогда раньше не сталкивались с Linux, не бойтесь: мы приготовили полное руководство для новичка, а на диске есть обновлённые и подлатан-

популярными ОС вполне оправдывают это лишнее действие. Начнём с того, что его исходный код открыт, а значит, вы можете законно скачать Linux и установить на свой компьютер. Также он содержит тонны программ, а новые можно загрузить одним щелчком.

В отличие от проприетарных ОС, виды Linux (или дистрибутивы) очень податливы. Вы можете выкинуть любое из стандартных приложений или даже весь графический интерфейс и заменить его более подходящим для ваших нужд. От одиночных компонентов и до сложных пакетов, выбор — один из козырей Linux и его визитная карточка.

Более того, Linux не только совместим со всем новым оборудованием, но и способен дать вторую жизнь старому оборудованию, чьи дни славы уже позади. Вдобавок вы можете делать всё то же, что и на Windows. От просмотра онлайн-видео до запуска последних игр, Linux будет работать несколько не хуже, чем любая другая система.

«Вдобавок вы можете делать всё то же, что и на Windows.»

ные Fedora 21 и Ubuntu 14.10. Это live-версии, их можно запустить прямо с DVD и сразу их опробовать.

Одна из самых больших помех для широкого распространения Linux — то, что он не прилагается к компьютеру в каждом магазине, и заполнить его чуть сложнее. Но преимущества Linux перед другими



Ubuntu

Шире всего известен дистрибутив Linux. Ubuntu уделяет особое внимание удобству использования рабочей среды. За 10 лет существования Ubuntu дал толчок разработке настольного Linux и является основным целевым дистрибутивом для сторонних разработчиков и производителей, желающих совместности с Linux.



Fedora

Предложение с открытым кодом от Red Hat. Fedora известна тем, что адаптирует и предлагает пользователям новые технологии и ПО. С годами дистрибутив сумел найти разумный баланс между инновациями и стабильностью, что делает его популярным одновременно среди новых и опытных пользователей Linux.



Mageia

Хотя на сегодня вышло всего четыре выпуска Mageia, родословная у её подхода к удобству начинается ещё в 1990-х годах. Mageia — общественный проект, который поддерживается некоммерческой организацией, управляемой советом избранных участников. Дистрибутив известен своими дружелюбными инструментами настройки и установки.

Установите Linux

Всё, что следует знать для вселения Linux на свой ПК.

Процесс установки Linux не так прост, но и не сложен. Точная последовательность шагов для установки каждого дистрибутива немного своя, но обычно графический установщик проводит вас через все необходимые шаги.

В принципе, установка Linux очень схожа с установкой программы, с некоторыми поправками:

» **Разбиение диска** В отличие от установки программы, установка Linux требует создания раздела на вашем диске. Это не проблема, если Linux будет единственной ОС на компьютере. Но если вы устанавливаете Linux рядом с другой ОС, например, с Windows, потребуется принять меры, чтобы сохранить существующие данные. Многие дистрибутивы Linux предлагают разбить диск автоматически, но легко и самим создать разделы из Windows с помощью инструмента «Управление дисками» (см. «Отводим место для Linux» внизу). Преимущество ручного разбиения в том, что вы сами решаете, сколько места выделить под Linux.

При разбиении диска не забудьте создать два новых раздела. Большой раздел, не менее 12 ГБ, нужен для самой ОС, его вы отформатируете в ext4. Также потребуется второй раздел — так называемый раздел подкачки [swap]. Если попросту, раздел подкачки дополняет физическую оперативную память на вашем компьютере. Общее правило для компьютеров с небольшим ОЗУ (1–2 ГБ): создайте раздел подкачки вдвое больший, чем ОЗУ компьютера. Для компьютеров с большим объемом ОЗУ лучше создать раздел подкачки такого же объема, как ОЗУ.

» **Защита данных** Во время установки многие дистрибутивы, включая Fedora и Ubuntu, позволяют зашифровать раздел с Linux. Эта опция даёт вам дополнительный слой безопасности, защищая ваши данные от неавторизованного доступа. Чтобы включить эту опцию, потребуется ввести пароль, которые впоследствии будет ключом для разблокировки данных.

Ещё один важный шаг установки — настройка учётной записи суперпользователя-root. У большинства дистрибутивов это часть

процесса создания пользователей, где вы указываете данные для обычного пользователя. Обычный пользователь не имеет права модифицировать систему, тогда как знание пароля root даёт возможность управлять системой.

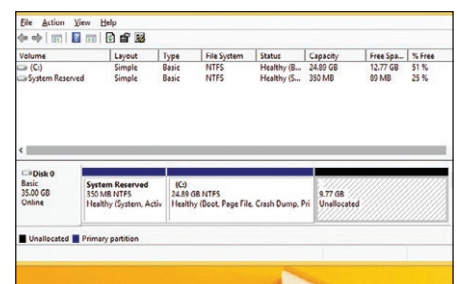
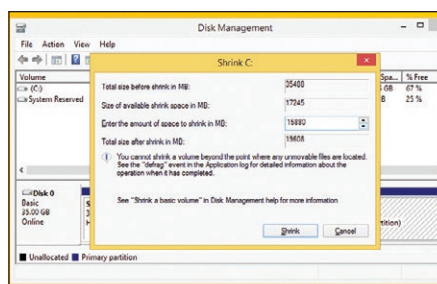
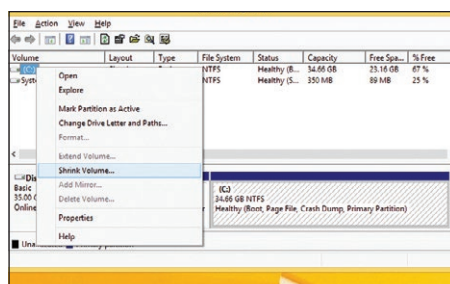
» **Двойная загрузка** Одна из вещей, о которых вам нужно знать при установке Linux — загрузчик. Это небольшая программа, которая сообщает компьютеру, где находятся различные ОС на диске. Большинство дистрибутивов Linux используют загрузчик *Grub 2*.

Обычно здесь вам не нужно ничего делать самим, даже когда вы устанавливаете Linux на компьютер с Windows 8, использующий UEFI BIOS с включённой опцией Secure Boot. Последние версии ведущих дистрибутивов, включая Ubuntu и Fedora, устанавливают UEFI-совместимый загрузчик, который будет корректно работать сразу. Однако, поскольку разные производители реализуют UEFI по-разному, есть вероятность, что после установки Linux вы не попадёте на экран *Grub*, а вместо этого сразу загрузитесь в Windows. В таком случае стоит попробовать включить режим совместимости [Legacy BIOS], в котором UEFI функционирует как обычный BIOS. Опция включения режима совместимости находится на экране настроек UEFI.

» **Проверка до установки** Почти все основные дистрибутивы, в том числе Ubuntu, Fedora и Mageia, дают возможность загрузиться в «живое [live]» окружение, которое позволит вам попробовать дистрибутив, не затрагивая содержимое вашего жёсткого диска. В живом окружении вы можете ознакомиться с дистрибутивом и проверить свое оборудование на совместимость с дистрибутивом.

Также заметьте, что дистрибутивы Linux распространяются как ISO-образы. Вы можете прожечь их на CD или DVD, в зависимости от их размера, используя функцию записи ISO-образов. Также можно записать ISO на USB-носитель. Для создания загрузочных USB-носителей существуют такие инструменты, как *Unetbootin* и *Yumi*, а Mageia рекомендует использовать утилиту *Rufus*. »

Отводим место для Linux: Меняем размер раздела Windows



1 Ужимаем Windows

Прежде чем переразбивать диск, потребуется уменьшить раздел с Windows, освободив место для нового раздела. Зайдите в Управление дисками [Disk Management] и нажмите правой кнопкой мыши на основном разделе (обычно ему присвоена буква C). Затем в контекстном меню выберите Сжать том [Shrink volume].

2 Создаём новый раздел

Диалог ужатия раздела покажет вам общее пространство на диске и максимально допустимый объём ужатия. Выжать из диска больше места, чем здесь объявлено, нельзя. Чтобы создать новый раздел, укажите его размер в мегабайтах и нажмите Сжать [Shrink] для запуска процесса.

3 Используем раздел

После завершения операции рядом с диском C: появится новый раздел с информацией о количестве свободного, или неразмеченного, места. Вы можете указать на это свободное место установщику дистрибутива Linux. Не забудьте повторить процесс, создав ещё один раздел для подкачки.

Различия дистрибутивов

Делаем первые шаги в Linux.

Одно из самых больших отличий, сбивающее с толку бывших пользователей проприетарных ОС, это отсутствие единого облика у настольного Linux. Рабочие столы по умолчанию в Ubuntu, Fedora и Mageia отличаются друг от друга. Мы говорим «по умолчанию», потому что, в отличие от проприетарных ОС, дистрибутив Linux позволяет удалить рабочую среду, заменив её на более подходящую к личным привычкам.

Идея рабочей среды (или рабочего стола) как отдельной от ОС сущности звучит непривычно для пользователей, переходящих с Windows или Mac. Но, как и всюду в Linux и свободном ПО, для внешнего вида дистрибутива у пользователя есть выбор.

Единство в многообразии

Дистрибутив Ubuntu использует собственную среду Unity [англ. Единство]. Самая заметная его часть — вертикальная панель Launcher, работающая как панель задач. Она содержит иконки для часто используемых приложений, которые вы можете изменять под ваши нужды. Также некоторые иконки имеют свои контекстные меню для быстрого доступа к часто требуемым функциям.

Первая иконка на Launcher открывает Dash — ответ Ubuntu традиционным системам меню. Внизу в Dash есть текстовое поле,

и всё, что вы там вводите, ищется среди приложений, документов, музыки, видео, мгновенных сообщений, контактов и другого контента. Более того, через Dash можно устанавливать и удалять приложения и видеть эскизы медиа-файлов. Также Unity включает Heads Up Display (HUD) — инновационный подход к меню приложений. С HUD вам больше не нужно искать функции, зарытые глубоко в меню. Просто нажмите клавишу Alt, чтобы открыть HUD из любого приложения, и выполняйте действия с помощью поискового поля, похожего на Dash.

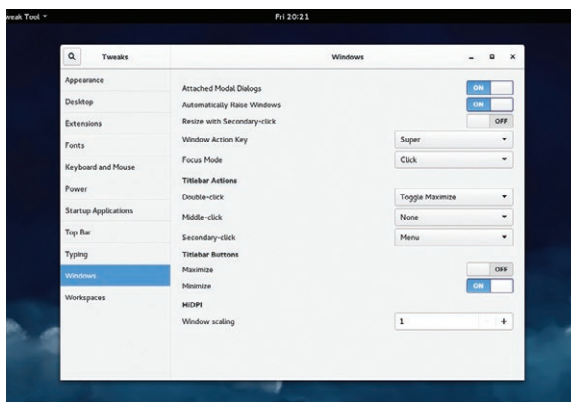
Поведение Unity по умолчанию — результат обширных исследований удобства пользователя в Canonical. Но вы найдёте несколько опций для изменения среды в Параметрах системы, доступных через иконку в Launcher — ту, что с шестерёнкой и гачечным ключом. Настройки сгруппированы по трём категориям. В категории Персональных есть настройки внешнего вида рабочего стола, с заменой обоев или изменением поведения панели. Обратите внимание на настройки Сетевых учётных записей, где можно подключиться к некоторым онлайн-сервисам, таким как Facebook и Google Docs, для интеграции их контента с приложениями. Так, добавление учётной записи Flickr интегрирует его с менеджером фотографий Shotwell.

Гном самолично

Gnome — ещё одна популярная рабочая среда, и Gnome 3 состоит примерно из тех же элементов, что и Unity в Ubuntu, но представляет их по-другому. Например, рабочий стол практически пуст. Щелкните по кнопке Обзор в левом верхнем углу, чтобы попасть на экран, очень похожий на Dash из Unity. На этом экране также содержится похожая на Launcher панель Избранное, для доступа к часто используемым приложениям.

В центре находятся эскизы всех открытых окон. Справа — переключатель рабочих пространств, который всегда показывает текущие рабочие пространства и одно дополнительное. При добавлении окна на второе пространство автоматически появится третье. Вверху расположена поисковая строка — она умеет искать любой текст среди приложений и локальных документов

➤ Пользователи дистрибутивов с Gnome могут изменить поведение своей рабочей среды через программу Gnome Tweak Tool.

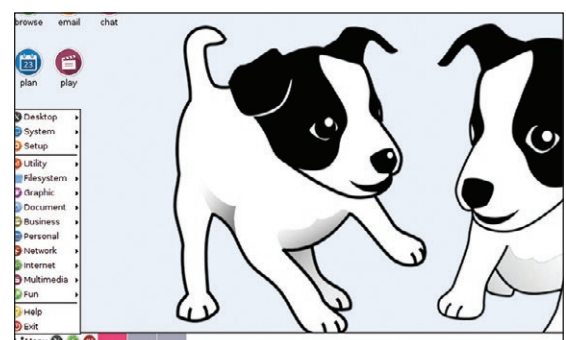


Воскрешаем старую рабочую лошадку

Одно из применений Linux состоит в том, что он позволяет вдохнуть новую жизнь в машины, которые не справляются с запросами современных ОС. Дистрибутивов для старого оборудования немало, но наш неизменный любимец — дистрибутив Puppy Linux.

Puppy применяет один из самых легких оконных менеджеров (JWM), и хотя может показаться неказистым внешне, способен превратить сонную старую клячу в скакового жеребца. Но главная причина звёздной производительности Puppy на оборудовании с ограниченными ресурсами — изобилие специальных лёгких приложений. Дистрибутив содержит графические

и офисные приложения, приложения для воспроизведения, редактирования и даже создания мультимедиа. Спецприложения Puppy умеют блокировать рекламу на сайтах, скачивать подкасты, совершать интернет-звонки, прожигать диски, и многое другое. Дистрибутив доступен в разных версиях. В Wary Puppy вошли более старое ядро и добавочные драйверы для поддержки периферии типа телефонных модемов. Есть варианты на базе недавних выпусков Ubuntu — такие как Tahrpup на Ubuntu 14.04, и Slacko на базе Slackware. Ядро в них новее, чем в Wary, но остаются преимущества приложений Puppy для слабых машин.



➤ У Puppy Linux очень полезный форум и много документации специально для новичков.

на компьютере, а также в онлайн-сервисах. Gnome содержит приложение Сетевые учётные записи, позволяющее подключаться к онлайн-сервисам вроде Google Docs и Flickr. Кстати, Fedora предложит вам подключить эти сервисы, когда вы загрузитесь в дистрибутив в первый раз.

Рабочий стол Gnome имеет также несколько собственных приложений, умеющих извлекать информацию и данные из добавленных учётных записей онлайн. Например, приложение Контакты Gnome собирает контакты из различных онлайн-источников, типа Gmail. Аналогично, Документы Gnome помогут находить документы из онлайн-репозиториях вроде Google Docs.

Новым пользователям стоит иметь в виду причуды Gnome. Во-первых, ни у одного окна в Gnome вы не найдёте кнопку Свернуть. Желая переключиться на другое приложение, откройте Обзор и запустите новое окно или выберите одно из уже открытых. Другой неочевидный момент — отсутствие каких-либо иконок на рабочем столе, а также возможности создать ярлыки или размещать папки на рабочем столе.

Зато Gnome позволяет настраивать себя: новые возможности добавляются буквально одним щелчком. Gnome поддерживает массу расширений-плагинов, и их можно подключать безо всякой установки. Просто зайдите на сайт расширений Gnome (extensions.gnome.org), найдите желаемый плагин и нажмите на кнопку, чтобы активировать его для своего рабочего стола. Некоторые популярные плагины разработаны для облегчения перехода пользователей с проприетарных ОС, таких как Windows.

Начните с KDE

В отличие от предыдущих двух сред, компоновка и поведение рабочей среды KDE и размещение его средства для запуска приложений Kickoff однозначно покажутся знакомыми пользователям других систем. Однако KDE так гибок, что многие дистрибутивы с KDE выглядят абсолютно непохожими. Во многом KDE является квинтэссенцией рабочего стола Linux, предлагая гибкость и мириады опций. Настройкам KDE буквально нет конца.

Одна из самых полезных возможностей KDE — Комнаты [Activities]. С её помощью вы можете создавать несколько контекстно-зависимых комнат, каждая из которых имеет собственный набор приложений и обстановки рабочего стола. Например, вы можете создать комнату для общения, где вы войдёте во все ваши учётные записи для мгновенного общения и будете видеть обновления из различных социальных сетей. Многие дистрибутивы KDE поставляются только с комнатой по умолчанию — Desktop Activity. Однако вы можете скачать комнаты из Интернета и построить их под свои привычки.

Более того, KDE имеет несколько интерфейсов, или Представлений [Views], созданных, чтобы выжать максимум из доступного

пространства рабочего стола. Для обычных экранов и нетбуков предлагаются разные Представления, но можно использовать любое Представление на компьютере любого типа. Чтобы изменить Представление, щёлкните правой кнопкой мыши на рабочем столе и в контекстном меню выберите Настройки рабочего стола по умолчанию. В открывшемся окне выберите вкладку Вид и попробуйте различные представления из выпадающего списка Тип.

Многие KDE-дистрибутивы размещают на рабочем столе виджет Просмотр папки — он отображает содержимое определённой папки в компактном окне, которое можно поместить где угодно на экране. Также существует представление Просмотр папки, в котором файлы и папки можно размещать где угодно на рабочем столе. Представление Поиск и запуск разработано для устройств с небольшими экранами и сенсорными экранами. У каждого представления есть добавочные настраиваемые элементы.

«Wayland и X сосуществуют на одной системе; устанавливайте и экспериментируйте.»

Настройки, доступные для отдельных элементов, также собраны в панели Параметры системы, наряду с общесистемными опциями для настройки дистрибутива, в котором запущен KDE. Это может запутать, но вам не обязательно устанавливать или проверять каждый параметр перед использованием рабочей среды. Настройка KDE — непрерывный процесс, а не разовая акция. Этот рабочий стол разработан так, чтобы расти и изменяться вместе с вашими требованиями к его использованию.

31 оттенок вкуса

Помимо этих трёх ведущих рабочих сред, существует множество других, которые вы можете установить в ваш дистрибутив. Есть и полноценные среды, такие как Cinnamon, и легковесные, такие как Xfce, LXDE и Mate. По сути, большинство известных дистрибутивов, включая Ubuntu, Fedora и Mageia, доступны в нескольких редакциях с разными рабочими столами.

Например, Ubuntu имеет несколько официально поддерживаемых версий. Среди них Kubuntu, который дополняет Ubuntu средой KDE, а также версия с Gnome, версия с Xfce и версия, использующая рабочий стол Mate. Дистрибутив Fedora, некогда известный как первый дистрибутив с Gnome, теперь имеет и чудесную KDE-версию. Точно так же можно использовать Mageia с Gnome. На самом деле, Mageia и Fedora также предлагают образы установочных DVD, позволяющих установить несколько рабочих столов.



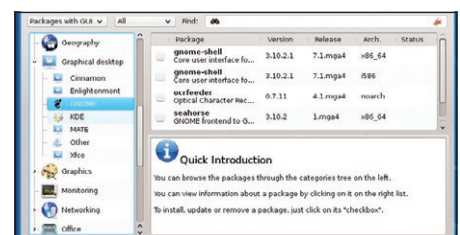
Меняем рабочую среду

Вы и сами можете установить в свой дистрибутив несколько рабочих сред, в том числе популярный рабочий стол Cinnamon. Он доступен в официальных репозиториях Fedora и Mageia, и устанавливается через соответствующие графические менеджеры пакетов.

В Ubuntu он доступен через PPA [ppa:gwendal-lebihan-dev/cinnamon-stable](https://launchpad.net/~ppa:gwendal-lebihan-dev/cinnamon-stable). Добавьте этот PPA в свою систему (как пояснено в основном тексте на стр. 45) и затем установите рабочий стол через Центр Приложений.

Установив несколько рабочих сред, вы легко можете переключаться между ними. Чтобы сделать это, выйдите из текущей среды и введите свои данные для входа в менеджере логина.

Перед входом, присмотритесь к кнопкам менеджера логина. Одна из кнопок откроет выпадающий список со всеми установленными рабочими средами. Выберите среду, которую хотите использовать, и менеджер логина впустит вас в неё. Таким способом можно попробовать их и решить, какая вам нравится больше всего. Выбор налицо!



» Через менеджер пакетов вашего дистрибутива можно установить несколько рабочих столов.

Установка программ

Расширяем выбранный дистрибутив.

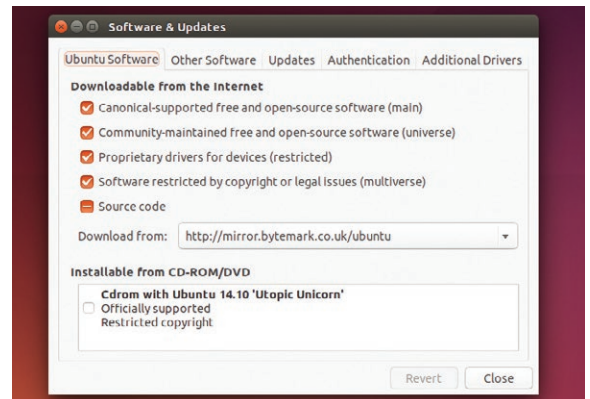
По сравнению с Windows, настольный дистрибутив Linux лучше подготовлен к работе сразу «из коробки». Средний дистрибутив Linux включает не примитивные приложения типа редактора простого текста или убогого инструмента для рисования, а полноценный офисный пакет и всеобъемлющий редактор графики. И это помимо основного набора приложений для стандартных задач, таких как web-сёрфинг, проверка почты, обмен мгновенными сообщениями с друзьями через различные соцсети, управление фотографиями, прослушивание музыки и просмотр видео.

Однако, так как все мы используем компьютеры по-разному, вам скорее всего понадобится какое-то ПО, не установленное по умолчанию. У вашего дистрибутива есть специальные инструменты, которые помогают устанавливать мириады качественных программ с открытым кодом без сложных мастеров установки.

Дистрибутивы Linux используют коллекцию инструментов, в графическом виде или в командной строке, которые все вместе называются системой управления пакетами. Эти инструменты помогают легко устанавливать, удалять и обновлять программы (называемые также пакетами). Отдельные программы группируются в пакеты. В дополнение к самой программе, пакет также содержит другую информацию: к примеру, список пакетов, или зависимостей, необходимых для нормальной работы программы. Более того, система управления пакетами использует базу данных, известную как репозиторий, чтобы следить за всеми доступными пакетами.

Азбука управления пакетами

Мир Linux в общем поделён между двумя форматами пакетов — RPM и Deb. Это прекомпилированные двоичные пакеты, призванные упростить установку для пользователей. RPM был создан в Red Hat Linux и используется в таких дистрибутивах, как Fedora и Mageia, а Deb используется в системах на базе Debian, например, в Ubuntu. Дополнительно, почти каждый из основных дистрибутивов содержит собственный набор графических инструментов, чтобы пользователь мог устанавливать, обновлять и удалять приложения. Вам также следует знать структуру репозитория дистрибутива и где он хранит программы.



► Помимо репозитория, можно выбрать зеркало для скачивания программ.

Ubuntu использует пакетную систему Advanced Packaging Tool, или APT. Вы можете использовать инструмент «Источники приложений», чтобы управлять репозиториями в Ubuntu. Инструмент показывает список репозиториях на четырёх различных вкладках. По умолчанию включены четыре официальных репозитория. Репозиторий Main включает официально поддерживаемое ПО, а Restricted включает ПО, недоступное под полностью свободными лицензиями. Два интересных репозитория — Universe и Multiverse, включающие соответственно пакеты, поддерживаемые сообществом, и несвободные пакеты.

В отличие от Ubuntu, Fedora использует систему управления пакетами RPM. Дистрибутив хранит репозитории в директории `/etc/yum.repos.d`; главный репозиторий называется `fedora.repo`.

Mageia использует пакет `urpmi`, это обёртка для системы RPM. У Mageia три официальных репозитория: `core` содержит пакеты с открытым кодом, `nonfree` — приложения с закрытым кодом, а репозиторий `tainted` — пакеты, способные нарушить патенты или законы об авторском праве в некоторых странах. Каждый из репозиториях поделён на четыре подраздела. `Release` включает стабильные пакеты, `Updates` — пакеты, обновленные после релиза дистрибутива, а `Backports` — пакеты новых

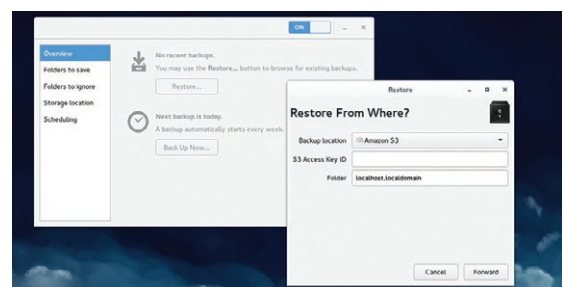
Руководство по резервному копированию

Ваш дистрибутив обязан включать инструмент для резервного копирования данных, и вам стоит ознакомиться с ним. Так, Ubuntu поставляется с приложением *Déjà Dup*, предназначенным для новичков. Его можно также установить в Fedora и Mageia.

Независимо от используемого инструмента, надо решить, что вы будете копировать и куда. Резервная копия всей домашней папки выглядит путём наименьшего труда, но обычно это перебор. Вместо этого стоит включить копирование папок внутри неё, таких как Загрузки и Документы. Также

проверьте важные приложения, такие как почтовые клиенты, которые хранят загруженные письма, вложения и адресную книгу внутри скрытых директорий (начинающихся с `./`) в домашней директории.

И ещё: от хранения резервных копий на другом разделе того же диска проку мало, ведь при отказе всего диска копия станет бесполезной. Одно из решений — хранить копии на отдельном диске или внешнем носителе. Либо, при хорошем интернет-канале, приложение поможет вам поместить резервные копии в облачном хранилище.



► Простой интерфейс *Déjà Dup* не отпугнёт даже тех, кто пользуется им в первый раз.

версий, перенесённые из репозитория Cauldron, который в будущем станет стабильным репозиторием для следующего релиза. Также есть репозиторий Testing, для тестирования программ пользователями.

Для настройки репозитория, запустите Центр управления Mageia и откройте Управление программами > Настройка источников установки/обновления ПО. Чтобы установить официальные онлайн-репозитории, нажмите Добавить и затем выберите либо Только источники с обновлениями, либо Полный набор источников. Первый вариант — минимальный для поддержания дистрибутива обновлённым, а второй позволяет устанавливать новые программы. Эти варианты заполнят окно списком репозитория. Поставьте галочку напротив репозитория, из которого хотите загружать ПО.

Пакеты с доставкой на дом

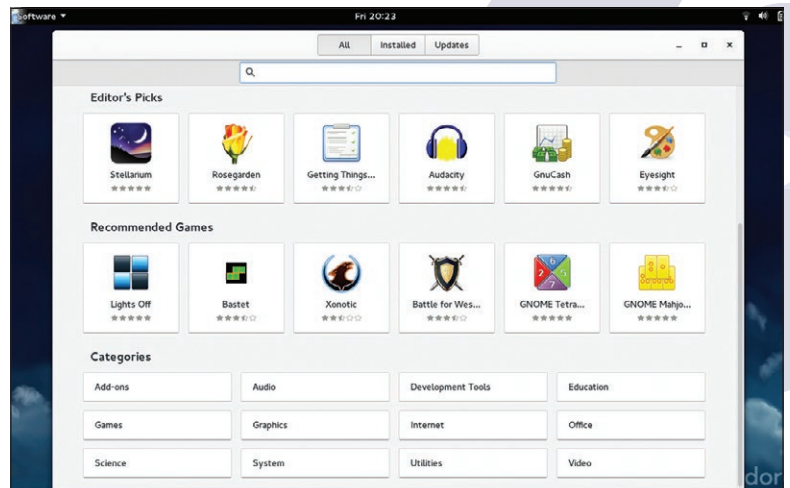
Все популярные дистрибутивы предлагают графические средства управления пакетами. Центр приложений в Ubuntu — один из лучших для этой задачи. Вы можете находить программы, щёлкая по категориям, отражающим нужный вам тип ПО. Выбрав категорию, вы увидите список приложений. Имеется и поисковое поле в правом верхнем углу, оно поможет найти программы по введённым ключевым словам. Найдя нужное приложение, нажмите кнопку Установить справа от него. Приложение и его зависимости загрузятся и автоматически установятся. Каждая свежееустановленная программа добавляется в Launcher; также её можно найти через Dash.

Fedora использует графический инструмент *PackageKit*, доступный через меню Gnome. Он тоже перечисляет программы по категориям и содержит поле для поиска. Найдя нужное ПО, нажмите кнопку Установить, и приложение добавится в вашу систему.

Графическое средство управления пакетами для Mageia называется *Drakrpm*. Оно менее симпатично, чем центры приложений в Ubuntu и Fedora, но очень функционально и достаточно интуитивно для использования. Можно фильтровать списки доступных приложений, чтобы видеть только пакеты с GUI, обновления безопасности, исправления ошибок и так далее. Группы приложений перечислены в боковой панели. Также есть поле поиска, чтобы разыскивать пакеты по ключевым словам. Найдя искомым пакет, просто отметьте соседнюю галочку и нажмите Применить.

Репо-люди

Более широкое сообщество открытого ПО способно предоставить гораздо больше пакетов по сравнению с официальными репозиториями, и почти каждый дистрибутив имеет механизм добавления и удаления программ из сторонних репозиториях.



Внешние репозитории в Ubuntu известны как Personal Package Archive [Личные архивы пакетов], или PPA. Вы можете добавить PPA-репозиторий в ваш дистрибутив с помощью средства «Источники приложений». Но сначала вам нужен адрес PPA. Он указан на странице PPA на сайте Launchpad и выглядит как-то напоподобие `ppa:example-ppa/example`. Теперь запустите средство и переключитесь на вкладку Другое ПО. Нажмите на кнопку Добавить и вставьте адрес PPA в открывшееся окно. После этого Ubuntu предложит вам обновить репозитории, чтобы PPA заработал.

» В большинстве дистрибутивов есть простой в использовании графический менеджер пакетов.

«По сравнению с Windows, настольный Linux лучше подготовлен к работе сразу.»

Точно так же и у Mageia есть сторонние репозитории, и для их добавления нужны URL. Заполучив URL, вызовите Центр управления Mageia, зайдите в Управление программами > Настройка источников установки/обновления ПО, щёлкните по опции Добавить источник. Во всплывшем окне введите адрес репозитория и его тип: HTTP или FTP.

Не обходится без сторонних репозиториях и Fedora; самый популярный из них — RPMFusion. Он разделён на два независимых репозитория, со свободным и несвободным ПО. Оба можно установить самостоятельно через браузер, следуя инструкциям на сайте соответствующих репозиториях (www.rpmfusion.org/configuration).

»

Грaбим Google

Все популярные программы от Google, включая Chrome, Планета Земля, аудио- и видеоплагин для Hangouts и другие можно установить в Linux. В официальных репозиториях ведущих дистрибутивов вы их не найдёте из-за лицензирования. Однако вы уже знаете всё, что нужно, чтобы легко установить их после небольшой подсказки.

Страница загрузок каждого приложения, поддерживаемого Google, содержит ссылки на 32-битную и 64-битную версии программы в форматах RPM и Deb. Скачайте пакет для своего дистрибутива

и дважды щёлкните на файле, чтобы установить его через менеджер пакетов. Все официальные пакеты приложений Google также устанавливаются в дистрибутив соответствующий репозиторий Google для обновления ПО.

Ещё одна популярная проприетарная программа, которую вы можете захотеть установить — Skype. Пользователи Ubuntu могут просто подключить репозиторий «Партнёры Canonical» в окне Источники приложений > Другое ПО. Это добавит официальные репозитории Skype, и вы сможете установить

программу через Центр приложений так же, как и любой другой пакет.

А вот Mageia включает Skype в репозитории non-free. Если вы подключили этот репозиторий, просто найдите пакет *get-skype*, который загрузит Skype с официального сайта. Вы также можете открыть страницу загрузок для Linux на сайте Skype, выбрать свой дистрибутив и архитектуру из выпадающего списка и загрузить файл Deb или RPM. Дважды щёлкните на этом файле, чтобы установить его через менеджер пакетов.

Мультимедиа-функции

Превратите свой дистрибутив в полноценный мультимедиа-центр.

Большинство настольных дистрибутивов Linux способны работать со всеми типами контента, который вы им предложите. Но контент, особенно большинство аудио и видео, иногда распространяется в закрытых форматах, отягощённых патентами. Дистрибутивы не могут воспроизводить такие файлы сразу после установки, однако в большинстве из них есть четкие процедуры установки компонентов для проигрывания несвободных форматов.

Ubuntu позволяет установить компоненты для ограниченных форматов файлов, таких как MP3, ещё во время установки. Если же вы уже установили дистрибутив, установите через менеджер пакетов *ubuntu-restricted-extras*, этот пакет включает популярные проприетарные кодеки и плагины.

В Fedora такие кодеки включены в сторонний репозиторий RPM Fusion. Вам нужно сперва подключить этот репозиторий, как мы описывали ранее, а затем запустить терминал и ввести следующие команды, чтобы скачать кодеки:

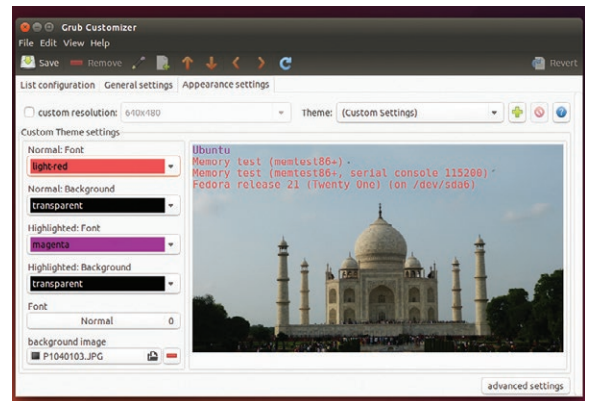
```
su -
yum install gstreamer{1,}-{plugin-crystalhd,ffmpeg,plugins-
{good,ugly,bad{,-free,-nonfree,-freeworld,-extras}{-extras}}
ffmpeg libmpg123 lame-libs
```

В Mageia мультимедиа-кодеки находятся в репозитории *tainted*, поэтому убедитесь, что он подключён, как мы описывали раньше. Затем откройте приложение *Welcome* из меню Утилиты и переключитесь на вкладку Приложения. Оттуда вы сможете установить несколько полезных и популярных пакетов, в том числе и кодеки.

Плюс роскоши

Несмотря на растущую популярность открытого формата WebM, много сайтов всё ещё используют для потоковых трансляций плагин Adobe Flash. Установка Flash замысловата, поскольку Adobe уже не разрабатывает Flash для *Firefox* на Linux. Единственный способ получить свежий Flash — использовать браузер Google *Chrome*, включающий плагин Flash на основе *Pepper*.

Если вы не хотите переходить на *Chrome*, можно установить старую версию Flash и продолжить пользоваться браузером



➤ Загрузчик Grub настраивается через *Grub-Customizer* (ppa:danielrichter2007/grub-customizer).

Firefox. Или вы можете извлечь последний *Pepper*-плагин Flash из *Chrome* и использовать его в открытом братце *Chrome* — браузере *Chromium*.

Пользователи Ubuntu могут установить Flash для *Firefox* командой

```
sudo apt-get install flashplugin-installer
```

Если вы используете *Chromium*, то последний *Pepper* Flash можно получить, установив пакет *pepperflashplugin-nonfree*.

Пользователи Fedora могут скачать плагин для *Firefox* с сайта Adobe, добавив его репозиторий. Если вы пользуетесь 64-битной версией, то команда

```
yum -y install http://linuxdownload.adobe.com/adobe-release/adobe-release-x86_64-1.0-1.noarch.rpm
```

скачает и установит правильный репозиторий. Вы можете импортировать ключи для репозитория:

```
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-adobe-linux
а уж потом устанавливать плагин:
```

```
yum -y install flash-plugin
```

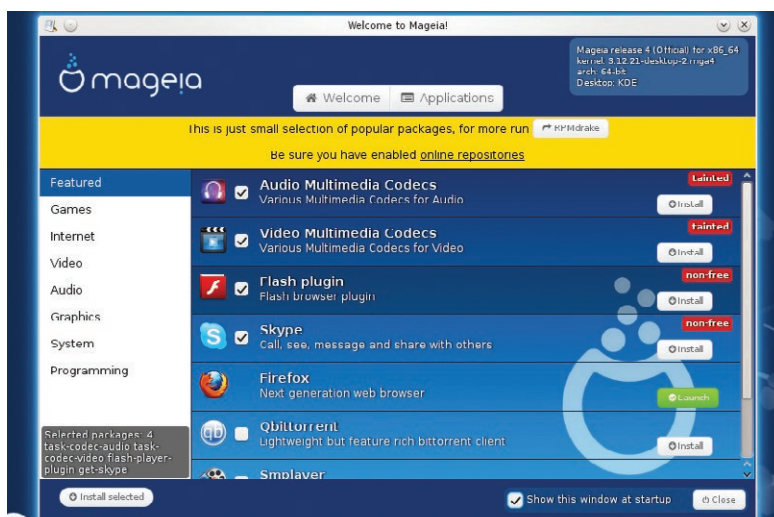
Пользователи Mageia могут просто подключить репозиторий *nonfree*, который содержит плагин Flash, и затем установить его из вкладки Приложения в *Welcome*.

Лучшие медиа-приложения

Большинство дистрибутивов содержит аудио- и видеопроигрыватели. Самые популярные — *Rhythmbox*, в дистрибутивах с *Gnome* идущий по умолчанию и хорошо интегрированный в *Ubuntu*, и стандартный плеер из *KDE* — *Amarok*. Помимо локальной музыки, оба умеют воспроизводить потоковое интернет-радио и подкасты. Если вы ищете плеер более привлекательного вида, запустите менеджер пакетов и осмотрите *Banshee* и *Clementine*.

Видеоплеер по умолчанию в большинстве *Gnome*-дистрибутивов — *Totem* (ныне именуемый просто *Videos*). Если вам нужно нечто более богатое функциями, попробуйте установить *MPlayer*. Это консольный плеер, но для него имеется несколько обёрток. Пользователи *Gnome* могут пользоваться *Gnome-Mplayer*, а пользователи *KDE* — *KMPlayer*. Есть также популярный кроссплатформенный плеер *VLC* с поддержкой практически всех форматов файлов.

➤ Приложение *Welcome* в Mageia — замечательное средство настройки дистрибутива для всех типов пользователей.



Сыграем?

Вперед на всех парах.

Игры долгое время считались ахиллесовой пятой Linux. За долгие годы мы накопили несколько качественных открытых игр, но у них не было той массовой популярности, как у популярных игр на проприетарных системах. Всё изменилось в начале 2013 года, когда Valve выпустила *Steam for Linux* — клиент для своего сверхпопулярного сервиса покупки игр.

Установить этот проприетарный клиент в ваш дистрибутив легко. В Ubuntu нужно подключить репозиторий «Партнёры Canonical» и установить клиент в *Центре приложений*. Пользователи Fedora должны установить репозитории RPM Fusion и установить *Steam* через менеджер пакетов. Аналогично, пользователи Mageia могут подключить репозиторий non-free и затем установить клиент *Steam* через приложение *Welcome*.

Но прежде чем запускать клиент *Steam*, убедитесь, что у вас есть необходимые драйверы для вашего графического оборудования. Это порой становится основной сложностью для пользователей, так как производители запрещают распространение компонентов своих закрытых драйверов. Для начала выясните марку и модель вашей графической карты, командой

```
lspci | grep VGA
```

Можно получить ещё больше информации, в том числе о тактовой частоте и возможностях:

```
sudo lshw -C video
```

Ваш дистрибутив позаботится, чтобы вы использовали самый подходящий открытый драйвер, как только вы в него загрузитесь. Среди пользователей Ubuntu популярен PPA Oibaf (ppa:oibaf/graphics-drivers), там имеются самые свежие открытые драйверы. Впрочем, пользователям «железа» от Nvidia и ATI/AMD следует для лучшей производительности в играх использовать проприетарные драйверы от соответствующего поставщика.

Пользователи Ubuntu могут получить последние стабильные драйверы Nvidia из PPA X-Swat (ppa:ubuntu-x-swat/x-updates). После его добавления загрузите драйверы командой

```
sudo apt-get install nvidia-current
```

Пользователи Fedora найдут последний драйвер GeForce в репозитории RPM Fusion. Добавив репозиторий, установите драйвер следующим образом:



```
yum install kmod-nvidia xorg-x11-drv-nvidia-libs kernel-devel acpid
```

Пользователям Mageia нужно сначала включить репозиторий non-free, а затем запустить Центр управления Mageia и открыть раздел Оборудование > Настройка графического сервера. В открывшемся окне щёлкните на переключателе рядом с надписью «Графические карты», чтобы увидеть список карт. Выберите свою. Если у Mageia есть проприетарный драйвер для этой карты, она его установит.

Карты AMD в списке Mageia присутствуют, но для Ubuntu или Fedora лучший источник проприетарных драйверов — сайт AMD (<http://support.amd.com/en-us/download>). Эта страница содержит несколько выпадающих списков, чтобы вы смогли выбрать драйвер именно для вашей графической карты. Затем скачайте предложенный драйвер и распакуйте его, чтобы увидеть скрипт `.run`. Перед установкой драйвера убедитесь, что у вас установлены необходимые зависимости:

```
sudo apt-get install dh-make dh-modaliases execstack libc6-i386 lib32gcc1
```

Сделав это, выполните скрипт

```
sh ./amd-driver-installer-13.35.1005-x86.x86_64.run
```

Запустится установщик проприетарного драйвера AMD Catalyst, который также установит программу для управления GPU Catalyst Control Center. По завершении работы установщика вернуться в терминал и введите

```
/usr/bin/aticonfig --initial
```

чтобы выполнить настройку драйвера. **LXF**

► Дистрибутивы в своих репозиториях содержат открытые игры, а *Steam* предлагает доступ к играм класса AAA, таким как *Dying Light*.



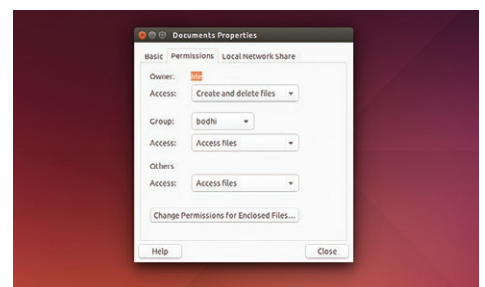
Основы управления пользователями

При установке Linux автоматически настраивается для одиночного пользователя, но легко добавить и другие учётные записи. Суперпользователь root имеет полный доступ к системе и её настройке; его дело — чисто административные задачи.

Обычные пользователи могут повысить свои привилегии через команды `su` и `sudo`. Пользователи объединены в группы и наследуют привилегии от них. Каждый пользователь Linux состоит хотя бы в одной группе. Поэтому можно контролировать, какие папки и файлы доступны пользователю или группе. По умолчанию, файлы пользователя доступны только для этого пользователя, а системные файлы доступны только

для root. В Linux на файлы и папки можно выдать права чтения, записи и запуска определённым пользователям. Это позволяет, например, иметь общий файл, который другие пользователи смогут читать, но не изменять.

Доступ к файлам контролируется через систему разрешений. Список разрешений вы получите от команды `ls -l`. Без указания имени файла она перечислит разрешения всех файлов в директории. Различные символы в выводе соответствуют праву на чтение (`r`), запись (`w`) и запуск (`x`) для владельца, его группы и всех остальных. Вы можете изменить разрешения для файла или папки командой `chmod` или через файловый менеджер.



► Файловый менеджер практически любого дистрибутива позволяет изменять группу, назначенную файлу или папке.



Михалис Цукалос объясняет всё, что вы хотели знать о NoSQL, но боялись спросить, и узнаёт, за что админы любят эту высокоскоростную систему.

Мир баз данных движется медленно (хотя, вероятно, не при транзакциях), и если происходит революция, то последствия ощущаются лет через десять. Придуманый в 1998 г. NoSQL — это базы данных, начавшие жизнь без употребления стандартного языка запросов SQL. Но что оказалось ещё более революционно, сами структуры баз данных отошли от стандартной реляционной модели ради скорости и простоты.

Как и говорит их название, базы данных (БД) NoSQL были изначально разработаны с целью отказа от SQL. Поладить с SQL они могут, но вместо этого используют различные другие языки запросов. Эти языки могли бы появиться и в 1998-м, но NoSQL не получил распространения до поздних нулевых, когда его выбрали как объединенный хэштэг в Твиттере для группы проектов нереляционных баз данных, ратовавших за малый размер и уникальность.

Если вы раздумываете о том, стоят ли БД NoSQL вашего внимания, помните, что в соответствии с рейтингом движков БД (<https://db-engines.com/en/ranking>), *MongoDB*, популярная база данных NoSQL, на данный момент является пятой по популярности после Oracle, *MySQL*, *SQL Server* от Microsoft и *PostgreSQL* — и даже Oracle имеет NoSQL-версию своей знаменитой БД.

«БД NoSQL — для Сети, и не поддерживают присоединения, сложные транзакции...»

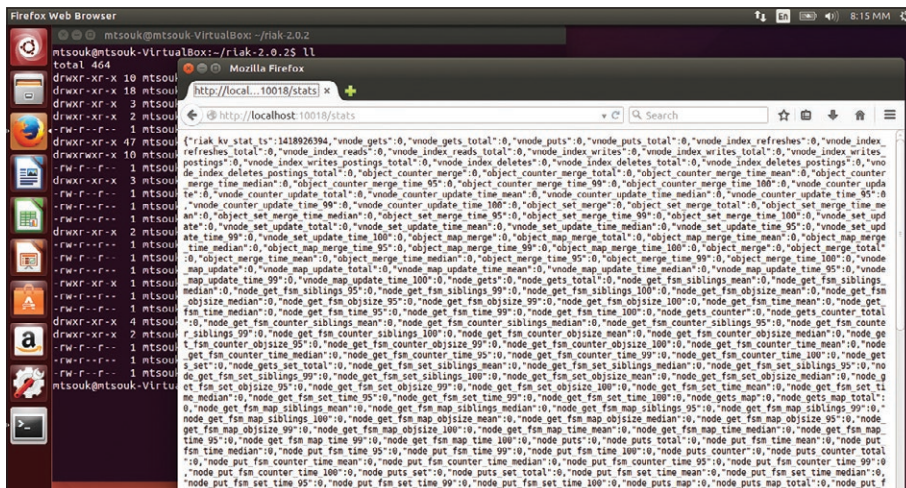
Проблема реляционных баз данных в том, что хранения сложной информации нужно разложить её на кусочки и поля и разместить во множестве всяких таблиц. Аналогично, чтобы добыть данные, придётся извлечь все эти кусочки и поля и собрать их снова. Обе эти задачи не слишком эффективны, особенно если у вас большой и загруженный

сайт, который постоянно сохраняет и запрашивает данные.

Следующий шаг — использовать для запуска вашей базы данных несколько машин, но это также создаёт проблему, потому что реляционные базы данных изначально создавались для запуска на одиночных системах. Поэтому крупные компании, такие как Google и Amazon, разрабо-

тали собственные базы данных, *Bigtable* и *Dynamo* соответственно, довольно сильно отличающиеся от традиционных реляционных БД; они то и вдохновили движение NoSQL. Что собой представ-

ляет база данных NoSQL, определить достаточно сложно, но вы можете выделить несколько общих признаков таковой: она не реляционная; с открытым кодом (правда, не всегда); не имеет схемы; легко распространяется на несколько машин (и тоже не всегда); а также пытается обслуживать данные web-культуры XXI века.



У каждого сервера *Riak* есть web-интерфейс. Здесь мы получаем доступ к статистике сервера, используя <http://localhost:10018/stats/>. Номер порта и IP определяются в *riak.conf*.

Итак, базы данных NoSQL разработаны для Сети и не поддерживают присоединения, сложные транзакции и другие особенности языка SQL. Их терминология тоже слегка отличается, так что давайте перейдём к деталям.

Главное преимущество БД NoSQL — пригодность для работы с большими объемами данных и с web-приложениями в реальном времени. Они также предлагают удобное масштабирование и позволяют безболезненно реализовать высокую доступность. Ими легче управлять, а также настраивать их и запускать, и они могут хранить сложные объекты. Вдобавок, гораздо проще разработать приложение для NoSQL, используя их же базы данных. Их схема легко меняется без всяких задержек, потому что, на самом деле, схемы у них нет. Большая их часть (за исключением NoSQL от Oracle) — проекты с открытым кодом.

К основным недостаткам баз данных NoSQL относят то, что они требуют абсолютно нового способа мышления, и даже в этом случае при работе с крупными и/или важными проектами вам всё равно понадобится DBA [Database administrator — администратор баз данных]. Если вашей компании нужны базы данных и SQL, и NoSQL, это означает две совершенно разные системы программирования и управления, так что вам понадобится больше людей. Будучи относительно новыми, базы данных NoSQL не столь зрелы, как реляционные, поэтому выбор базы данных NoSQL для критически важной задачи — не самый безопасный; но эта проблема через пару лет исчезнет. Последний недостаток такой: хотя кажется,

что схемы у них нет, для сколько-нибудь серьёзной работы с вашими данными придётся принять некую неявную схему. Это не столь неожиданно, потому что уйти от наличия схемы, пусть даже неформальной, при работе с данными не удастся.

Есть несколько разновидностей баз данных NoSQL — каждая хороша в какой-то одной или нескольких областях, но не во всех. Можно классифицировать базы данных NoSQL в соответствии с их моделями данных:

» **Документ (Document)** Очень распространённая модель данных. Она рассматривает базу данных как большое хранилище документов, где каждый документ — многосоставная структура данных, обычно представляемая в формате JSON. Но допускается хранить документы в любом желаемом формате. Типичные представители NoSQL-БД документов — *MongoDB*, *CouchDB* и *RavenDB*.

» **Ключ-значение [Key-Value]** Также распространённая модель данных, подобная структуре данных хэш-карты, где у вас есть ключи и вы просите базу данных вернуть значение, хранимое для данного ключа. Значение может быть любым — от одного числа до целого документа. База данных ничего не знает о хранящихся в ней данных. Примеры NoSQL-БД типа ключ-значение — *Riak*, *Redis* и *Project Voldemort*.

» **Столбец-семейство [Column-family]** Довольно сложная модель данных. У вас есть «ключ строки», позволяющий хранить и получать доступ к нескольким семействам столбцов. Каждое семейство столбцов — это комбинация сочетаемых друг с другом столбцов. Модель данных может

быть немного труднее остальных, но в результате сокращает время поиска. Примеры NoSQL-БД с семействами столбцов включают *Cassandra* и *Apache HBase*.

» **Граф [Graph]** Эта модель полностью отличается от предыдущих трёх, поскольку основана на структуре графов. Логично, что базы данных Graph NoSQL отлично обрабатывают иерархии и связи между чем-либо; однако обработка реляционных баз данных похожим образом представляется крайне трудной и затратной по времени задачей. Пример NoSQL-БД типа graph — *Neo4j*.

В данной статье за нашу тестовую NoSQL-БД мы примем *Riak*.

Установка Riak

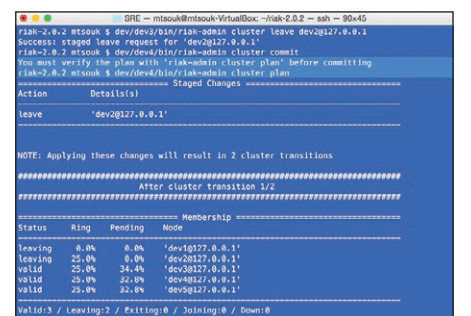
Первое, что вам следует знать перед установкой *Riak* — тут нужен установленный Erlang [см. Академия кодига LXF194, стр. 92]. Лучший способ установить *Riak* — скомпилировать его из ядра, тогда у вас будет лучше управляемая и полностью автономная сборка. Прodelайте следующие шаги:

```
$ wget http://s3.amazonaws.com/downloads.
  basho.com/riak/2.0/2.0.2/riak-2.0.2.tar.gz
$ tar xzvf riak-2.0.2.tar.gz
$ cd riak-2.0.2
$ make rel
```

Альтернатива — получить исходник ядра *Riak* из GitHub и скомпилировать его, как указано ранее:

```
$ git clone git://github.com/basho/riak.git
$ cd riak
$ make rel
```

Оба способа должны сработать без особых проблем; мы выбрали первый. Успешно скомпилировав *Riak*, вы можете найти его главные двоичные файлы в каталоге `./rel/riak/bin`. В том же каталоге, где вы собрали *Riak*, вы можете запустить `make devrel` и получить восемь готовых к запуску баз данных *Riak*, их мы будем использовать



» Создать кластер *Riak* с пятью узлами довольно просто — см. подробности на стр. 51.

Раскидать и сократить

MapReduce — это продвинутая техника запроса и инструмент для сбора данных, используемый в базах данных NoSQL. Это альтернативная техника запроса базы данных, весьма несхожая с обычными декларативными. Вы даёте базе данных инструкцию, как найти требуемые вам данные, и MapReduce пытается их искать (на стр. 50 вверху — простой

пример того, как работает MapReduce). Иногда использование MapReduce бывает довольно запутанным. Тем не менее, это позволяет создавать запросы, которые было бы крайне сложно создать на традиционном SQL.

Поняв принцип MapReduce и попрактиковавшись в нём, вы найдёте его очень надёжным и удобным.

Решение MapReduce отнимает больше времени на реализацию, но лучше расширяемо, чем решение SQL. Оно придаёт некую гибкость, которая на данный момент недоступна в процессе агрегирования. Самое сложное — решить, стоит ли применять технику MapReduce для случая ваших конкретных проблем. Это знание приходит с опытом!

для примера. Это главная причина получить код ядра *Riak* и скомпилировать его для себя.

Прежде чем продолжить нашу статью, познакомим вас с некоторыми терминами. Во-первых, узел [node] *Riak* аналогичен физическому серверу. Кластер *Riak* — это 160-битное пространство, разбиваемое на равные по размеру разделы. В свою очередь, разделы — это части, на которые делится кластер *Riak*. За раздел в *Riak* отвечает vnode. Vnode'ы координируют запросы для разделов, контролируемых ими. Кластер *Riak* может иметь множество узлов, которые находятся на одних и тех же или разных физических машинах. Кольцо [Ring] — это 160-битное пространство, разбитое на равные разделы, а участок [bucket, «ведро»] — это имена для хранящихся в *Riak* данных. Внутри *Riak* вычисляет 160-битный двоичный хэш каждого участка/ключа и помещает это значение в задаваемое кольцо всех подобных значений.

Как вы увидите далее в этой статье, любой интерфейс клиента для *Riak* взаимодействует с объектами в категориях «участок» и «ключ», в котором хранится некоторая величина, а также тип участка, определяющий его свойства.

По умолчанию в *Riak* задан режим работы кластера, состоящего из нескольких узлов. Узлы *Riak* по умолчанию не являются клонами друг друга.

Вы можете запустить три экземпляра сервера *Riak* — все восемь серверов запускать не надо — выполнив следующие команды:

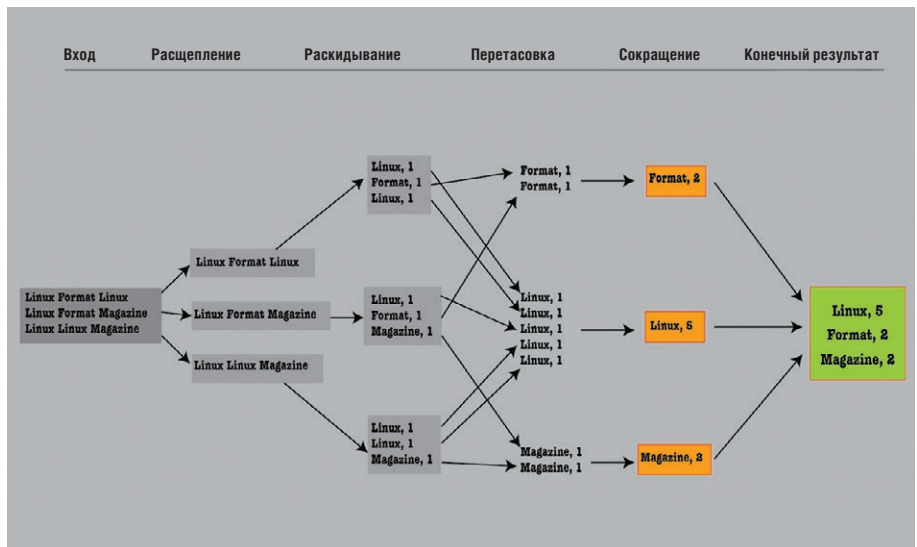
```

$ ./dev/dev1/bin/riak start
$ ./dev/dev2/bin/riak start
$ ./dev/dev3/bin/riak start
$ ./dev/dev1/bin/riak start
Node is already running [Узел уже запущен!]
    
```

Каждый сервер *Riak* предлагает web-интерфейс (на стр. 49 вверху — пример того, что вы увидите, подключившись к серверу *Riak*). Номер порта и IP-адрес сервера определяются в файле **riak.conf**. Это обычный текстовый файл, и его можно редактировать. Следующая команда показывает IP и номер порта, управляющий каждым сервером *Riak*:

```

$ grep listener.http.internal `find ./dev -name riak.conf`
./dev/dev2/etc/riak.conf:listener.http.internal = 127.0.0.1:10028
./dev/dev1/etc/riak.conf:listener.http.internal = 127.0.0.1:10018
    
```



➤ **Пример MapReduce.** Возможно, выглядит простецки, но MapReduce — очень мощная технология. Попытка сделать то же самое с помощью SQL способна оказаться крайне сложной.

```

./dev/dev3/etc/riak.conf:listener.http.internal = 127.0.0.1:10038
    
```

И так далее... с каждым узлом *Riak* связано имя. Имя можно изменить путём изменения переменной имени узла в файле **riak.conf**. Первый сервер (dev1) использует порт под номером 10018, второй сервер (dev2) — под номером 10028, а третий (dev3) — под номером 10038. Версии *Riak*, предшествующие 2.0, использовали файл конфигурации **app.config**, потом его заменили на **riak.conf**.

Простейший способ узнать, запущен узел *Riak* или нет — через команду **curl** и web-интерфейс узла, пропинговав его:

```

$ curl http://localhost:10018/ping
OK
$ curl http://localhost:10038/ping
curl: (7) Failed to connect to localhost port 10038: Connection refused
[Не удалось соединиться ... Отказ в соединении]
Альтернатива — следующие команды:
$ ./dev/dev1/bin/riak ping
pong
$ ./dev/dev6/bin/riak ping
Node 'dev6@127.0.0.1' not responding to pings.
[Узел ... не отвечает]
    
```

Преимущество «способа **curl**» в том, что он применим и с удалённой машины — при условии,

что сервер *Riak* также слушает внешний IP-адрес — не входя в машину, на которой запущен узел *Riak*. Сервер *Riak* dev1 можно остановить командой **./dev/dev1/bin/riak stop**.

Riak использует **epmd** — Erlang Port Mapper server [Сервер поиска портов] — который играет ключевую роль во всей работе *Riak*. Работа **epmd** начинается автоматически командой **erl**, если узел должен быть распределен и нет работающих экземпляров. Работа **epmd** позволяет узлам *Riak* находить друг друга. Это очень лёгкий и безвредный процесс, способный продолжать работу даже после остановки всех узлов *Riak*. Его можно прекратить вручную, когда вы прекратите работу всех узлов *Riak*, но это необязательно. Команда ниже показывает все имена с зафиксированным в данный момент работающим процессом **epmd**:

```

$ epmd -names
epmd: up and running on port 4369 with data:
name dev3 at port 49136
name dev1 at port 55224
name dev2 at port 48829
    
```

Хранение и извлечение данных

Вы можете подключиться к серверу *Riak* dev1 и сохранить документ, используя web-интерфейс:

Сравнительный анализ Riak

Basho (<http://basho.com>) предлагает инструмент сравнительного анализа для *Riak*, написанный на Erlang. Вы можете получить и установить его командами

```

$ git clone git://github.com/basho/basho_bench.git
$ cd basho_bench
$ make
    
```

Затем следует запустить **./basho_bench myconfig.config**, чтобы получить данные сбора инструмента, или либо самому создать **myconfig.config**, либо изменить уже существующий.

Существующие файлы находятся в каталоге **examples**. Мы использовали файл **examples/basho_bench_ets.config** в качестве начальной точки и добавили строку **riakclient_nodes,['dev1@127.0.0.1','dev2@127.0.0.1']**.

Basho Bench создаёт один процесс Stats и рабочие процессы на основе того, что определено в параллельно действующем варианте конфигурации в файле **myconfig.config**. Как только эти процессы создадутся и запустятся, Basho Bench пошлет команду **run** всем рабочим процессам,

запуская тестирование. Процесс Stats получает уведомление по завершении любой операции. Кроме того, он получает сведения о времени, потраченном завершившейся операцией, и сохраняет эти данные в гистограмме.

Все результаты находятся в каталоге **tests**. Последние результаты могут быть обнаружены через символическую ссылку **./tests/current/**. Для создания графика по текущим результатам запустите **make results**. [См. стр. 51 внизу для примера вывода].

```
$ curl -v -X PUT http://127.0.0.1:10018/riak/LXF/
test -H "Content-Type: text/html" -d "<html>
<body><h1>This is a test.</h1></body></html>"
```

Всё, что действительно хранится в тестовой зоне **/riak/LXF** — это то, что следует за опцией **-d**. Когда вы успешно добавите новое значение, *Riak* вернёт код HTTP 204. Как вы уже знаете, *Riak* — хранилище, работающее по принципу ключ–значение, и чтобы извлечь необходимое значение, нужно предоставить *Riak* ключ. Вы можете подключиться к серверу *Riak* dev1 и запросить ранее сохранённый документ, перейдя по URL <http://127.0.0.1:10018/riak/LXF/test>. Каждый URL следует шаблону <http://SERVER:PORT/riak/BUCKET/KEY>.

Вот команда, которая возвращает список доступных участков:

```
$ curl -i 'http://127.0.0.1:10018/riak?buckets=true'
HTTP/1.1 200 OK
Vary: Accept-Encoding
Server: MochiWeb/1.1 WebMachine/1.10.5
Date: Fri, 19 Dec 2014 21:13:37 GMT
Content-Type: application/json
Content-Length: 33
{"buckets":["LXF","linuxformat"]}
```

А следующая команда вернёт список ключей в участке:

```
$ curl 'http://127.0.0.1:10018/buckets/LXF/keys?keys=true'
{"keys":["test2","test","test3"]}
```

В большинстве случаев, чтобы получить доступ к базе данных *Riak*, вы будете использовать скрипт, написанный на каком-нибудь языке программирования.

Ниже приведён скрипт Python, который соединяется с базой данных *Riak*, сохраняет и извлекает документ:

```
import riak
# Соединение с кластером
client = riak.RiakClient(pb_port=10017,
protocol='pbc')
# Имя участка
bucket = client.bucket('python')
# "myData" - имя используемого ключа
aRecord = bucket.new('myData', data={
'Имя': "Михалис",
'Фамилия': "Цукалос"
})
# Сохраняем запись
```

```
aRecord.store()
# Определяем ключ для извлечения записи
myRecord = bucket.get('myData')
# Извлекаем запись!
dictRecord = myRecord.data
# Распечатаем, чтобы убедиться в
правильности.
print dictRecord
$ python myRiak.py
{'Фамилия': u'Цукалос', u'Имя': u'Михалис'}
```

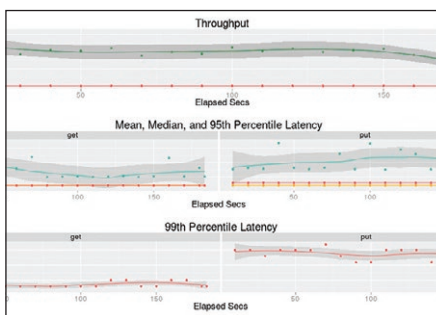
Значение `pb_port 10017` определено в файле `./dev/dev1/etc/riak.conf` параметром `listener.protobuf.internal`. Это порт Protocol Buffers [язык описания данных, предложенный Google как альтернатива XML, — *прим. пер.*], используемый для соединения с кластером *Riak*.

Вследствие гибкости способов хранения данных у баз данных NoSQL, установка, запрос и обновление базы данных NoSQL более сложные, чем у баз данных, использующей SQL.

Создание кластера Riak

Создавать кластеры в *Riak* и управлять ими относительно просто с помощью команды **riak-admin**. Если вы пытаетесь добавить к кластеру еще не запущенный узел, вы потерпите неудачу и получите следующее сообщение об ошибке:

```
$ dev/dev2/bin/riak-admin cluster join
dev1@127.0.0.1
Node is not running!
$ ./dev/dev2/bin/riak start
$ dev/dev2/bin/riak-admin cluster join
dev1@127.0.0.1
Success: staged join request for 'dev2@127.0.0.1'
to 'dev1@127.0.0.1'
```



► *Riak* предлагает инструмент сравнительного анализа Basho Bench. График создан с помощью R.

```
[Успех: выполнен запрос join]
$ dev/dev2/bin/riak-admin cluster join
dev1@127.0.0.1
Failed: This node is already a member of a Cluster
[Ошибка: этот узел уже в каком-то кластере]
```

Таким же образом, попытавшись присоединить узел к самому себе, вы получите сообщение об ошибке:

```
$ dev/dev1/bin/riak-admin cluster join
dev1@127.0.0.1
Failed: This node cannot join itself in a Cluster
[Ошибка: этот узел не может присоединиться сам к себе в кластере]
```

Следующая команда показывает участников существующего кластера:

```
$ dev/dev2/bin/riak-admin status | grep members
ring_members : ['dev1@127.0.0.1','d
ev2@127.0.0.1']
$ dev/dev1/bin/riak-admin status | grep members
ring_members : ['dev1@127.0.0.1','d
ev2@127.0.0.1']
$ dev/dev3/bin/riak-admin status | grep members
Node is not running [Узел не запущен!]
```

Другая полезная команда, показывающая статус узлов:

```
$ ./dev/dev1/bin/riak-admin member-status
```

Статус присоединения является временным, а действительным он станет, когда все изменения, ожидающие своей очереди, будут применены. Если вам требуется, чтобы изменения вступили в силу, выполните команду **riak-admin cluster commit**.

Если вы снова выполните команду **riak-admin member-status**, вы увидите новый статус узла dev3, и команда **riak-admin cluster plan** отобразит изменения, которые вот-вот вступят в силу.

Чтобы узел действительно покинул кластер (см. стр. 49 внизу — там вы увидите, как выглядит взаимодействие кластера с пятью узлами), вы должны вначале рассмотреть изменения, используя команду **riak-admin cluster plan**, и затем принять их с помощью команды **riak-admin cluster commit**.

До сих пор при работе с базой данных *Riak* вы не видели никакого попечения о безопасности. Тем не менее, *Riak* поддерживает имена пользователей и пароли. Вы можете узнать больше о том, как *Riak* справляется с аутентификацией и авторизацией, на <http://bit.ly/RiakDocsAuthz>. **LXF**

Состоятельность данных

Состоятельность информации в базах данных критически важна. ACID (Atomicity [Атомарность], Consistency [Состоятельность], Isolation [Изоляция] и Durability [Прочность]) — это набор свойств, гарантирующих, что транзакция базы данных пройдёт надёжно. Атомарность означает, что сделанное в базе данных изменение должно работать или не работать как единое целое. Изоляция означает, что если на тех же данных в одно и то же время происходят разные вещи, недоделанные данные не должны быть видны. Прочность обеспечивает

то, что если пользователь уведомлен об успехе транзакции, она должна сохраниться и не будет отменена, даже если оборудование или программа впоследствии сломаются.

Базы данных типа Graph выполняют транзакции ACID по умолчанию, что не может не радовать. С другой стороны, не каждая проблема требует «идеального» исполнения ACID. *MongoDB* является ACID-совместимым на уровне одного документа, но не поддерживает обновления нескольких документов, которые могут отмениться. Иногда терять

транзакцию или иметь временно несогласованную базу данных в обмен на скорость — это нормально.

Следует дотошно проверять характеристики баз данных NoSQL и решать, подходят ли они вашим требованиям. Тем не менее, если состоятельность данных для вас абсолютный приоритет, вы всегда можете обеспечить её посредством кода, если она не полностью поддерживается вашей базой данных NoSQL. Помните, что это может оказаться весьма нетривиальным, особенно в распределённых средах.



Fedora 21

Джонни Бидвелл надевает фетровую шляпу (правда, хлыст не берет) и ищет сокровища нового дистрибутива.

Мы знакомились с Fedora 20 **этак с год назад** [Обзоры LXF180, стр. 15], и вот уже наступает черед нового свежего релиза этого

почтенного дистрибутива, который изготовлен в трех разных вариантах.

Когда вы будете читать этот наш материал, Fedora 21 уже выйдет в большой мир и будет кочевать в Интернете. Кроме того, на момент написания, так как релиз несколько раз откладывался для устранения недочетов, автору пришлось довольствоваться последним релиз-кандидатом (подчиняясь сроку

сдачи статьи, а также коварным соблазнам бара на корпоративе). Поэтому мы не будем упоминать некоторые мелкие ошибки, которые наверняка уже будут исправлены к появлению официаль-

Корова], Beefy Miracle [Мясистое Чудо]; но если вы гонитесь только за эксцентрическими названиями, для вас всегда есть Ubuntu. Fedora, как и Debian, всегда ориентировалась на средний уровень

и выше. Это означает не сложность в работе, а явное отсутствие интерфейсов-мастеров для задач, удобнее всего выполняемых в командной строке. Fedora

«Отсутствуют интерфейсы-мастера для задач, удобнее выполняемых в командной строке.»

по умолчанию всегда включала SELinux, но если у вас нет каких-то особых требований, он не будет путаться у вас под ногами. Кстати, таким он был не всегда.

ного релиза. В нарушение традиции, данный релиз не носит своеобразного кодового имени, как его предшественники — Heisenbug, Schrodinger's Cat [Кот Шредингера], Spherical Cow [Сферическая

по умолчанию всегда включала SELinux, но если у вас нет каких-то особых требований, он не будет путаться у вас под ногами. Кстати, таким он был не всегда.

Мы могли бы целиком посвятить статью обзору всех обновлений и тщательному перечислению всех пакетов, присутствующих в установке по умолчанию этой новой Fedora, но делать этого не станем.

Для большинства пользователей все будет достаточно современно, кроме обычной возни с пакетами официальных проприетарных драйверов — но для них существуют способы и средства улучшения ситуации.

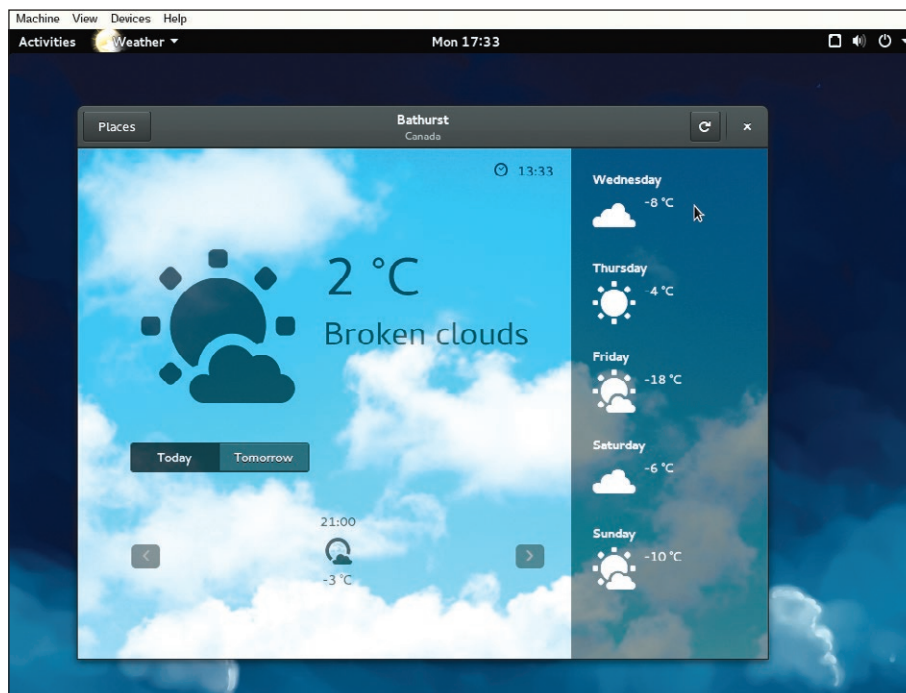
В общем, если вам нужен самый-самый свежий вариант, берите дистрибутив с возобновляемым релизом, например, Fedora Rawhide. Fedora — последний из основных дистрибутивов, чей рабочий стол по умолчанию — многими परिцаемый Gnome Shell; однако не переживайте, если вас это не устраивает. Здесь множество других рабочих столов, доступных для установки, и в столь именитом дистрибутиве пакеты обеспечивают достаточную простоту этого процесса. Самой сложной его частью станет заваривание чашечки чая, пока нужные пакеты скачиваются и устанавливаются.

Три — магическое число

Итак, о чем же стоит упомянуть? Ну, для начинающих — о том, что на самом деле это не один дистрибутив, а три: Fedora разделилась на разновидности Cloud, Server и Workstation. Были приложены колоссальные усилия, чтобы сделать образ Cloud как можно меньше — в частности, пакет *kernel* теперь является мета-пакетом: все модули, которые не относятся к минимальному образу Cloud, вынесены в отдельный пакет под названием *kernel-modules*.

Пакет *kernel-core*, помимо самого образа ядра, содержит только те модули, которые нужны для работы в качестве, например, EC2 или экземпляра OpenStack. Поэтому образ Cloud использует только пакет *kernel-core*, а образы Server и Workstation зависят от мета-пакета (который устанавливает пакеты *-core* и *-modules*).

В «облачном» релизе имеется также среда Red Hat Project Atomic, которая позволяет настроить так называемый Atomic Host: это легковесная платформа, чья единственная цель — размещение контейнеров Docker. Помимо Fedora 21, Atomic hosts также доступны из CentOS и Red Hat Enterprise Linux. Это амбициозный проект (он вполне согласуется с лозунгом Red Hat «вести, а не догонять»), и он напрямую соперничает с уже устоявшимся дистрибутивом CoreOS (который,



➤ Программа прогноза погоды не распознала Бат в Великобритании. Несомненно, Батерст в Канаде — отличный город, но нам больше нравится у себя.

между прочим, планирует раскол в среде контейнеров из-за введения конкурирующего формата App Container). Проект Atomic полностью отказался от традиционного управления пакетами — основная ОС обновляется с помощью новой технологии под названием *OSTree*, во многом подобно репозиториям *Git*, что облегчает задачу поэтапного обновления и возврата к исходному состоянию в случае проблем.

Со стороны сервера появилось несколько новых технологий для оптимизации планирования и администрирования. Rolekit предоставляет систему внедрения, которая позволяет быстро оснастить сервер для выполнения по запросу определенной функции или роли. Каждой ролью можно управлять из того же интерфейса, обеспечивая простоту через единообразие. Cockpit — это сетевой интерфейс управления сервером, который будет особенно привлекателен для тех, кто только начал заниматься администрированием в Linux. Как и панель управления Webmin, Cockpit позволяет запускать и останавливать сервисы, проверять лог-файлы и выполнять всю механическую работу без необходимости вводить какие-либо команды (и где тут счастье?..). Cockpit может также управ-

лять удаленными серверами, а самое лучшее — то, что он не навязывается вам: сервисы, запущенные из Cockpit, можно остановить из командной строки, и наоборот.

Вероятно, самой амбициозным ходом Fedora Server 21 является ввод OpenLMI, Linux Management Infrastructure, цель которой — абстрагировать несметное число инструментов, синтаксисов и особенностей, с которыми раньше приходилось знакомиться и впоследствии учитывать системным администраторам. OpenLMI предлагает стандартизованный API, с доступом из C/C++, Python, Java или старомодных инструментов командной строки, позволяя с легкостью справляться с обычными задачами. OpenLMI предлагает интерфейс клиент-сервер и, таким образом, становится доступным для удаленного управления и мониторинга. Имеется также и web-интерфейс, и, как таковой, имеет множество пересечений с вышеупомянутым Cockpit. Однако эти инструменты отнюдь не наступают друг другу на ноги, и наличие разных возможностей очень радует. Потенциал OpenLMI гораздо шире, чем Fedora, он использует стандарты Distributed Management Task Force для перемещения данных и работает в тандеме

Обновление

Если вы уже являетесь пользователем Fedora и хотите быстро и эффективно обновить свою систему, никак не меняя своих тщательнейшим образом подобранных пакетов, то для вас есть хорошая новость: вы можете использовать инструмент *FedUp*, который именно это и делает. Проверьте, что все обновлено, затем установите *FedUp*:

```
$ sudo yum update
```

```
$ sudo yum install fedup
```

И хотя за источник обновления можно выбрать ISO *FedUp* (используя опцию *--iso*), проще найти всё в Интернете, поскольку в общем пакеты с образами меняются очень быстро; команда

```
$ sudo fedup-cli --network 21 --product=nonproduct
```

найдет все нужные пакеты, сделает перезагрузку и потом установит их без всякого вмешательства

пользователя в дальнейшем. Если вы хотите, чтобы в вашей системе было все то, что есть, например, в релизе Workstation, можете вместо этого просто указать *--product=workstation*.

Помните, что в дополнение к другим рабочим столам, которые вы используете, у вас будет установлен Gnome, так что, возможно, позднее вы решите навести порядок.



со стандартными программами системы (не подменяя их собой). Те, кто уже знаком с Argonaut, возможно, не найдут применения для OpenLMI (ну разве что в качестве объекта насмешек), однако для тех, кто только вступает в игру, он определенно облегчит процесс обучения.

Итак, перейдем в редакции Desktop, oh, простите, Workstation, в которой мы... не обнаружили особых сюрпризов. Процесс установки прост, Btrfs не является файловой системой по умолчанию (эта честь выпала Ext4, и так, вероятно, и останется вплоть до появления Fedora 23), однако ее можно поставить всего парой щелчков мыши, если вы так жаждете COW (Copy-on-Write) [см. «Вот она — техника-2015», LXF194, стр. 30]. Если вас устраивают настройки по умолчанию, то вы получите три раздела: boot, root и swap, и два последних будут управляться через LVM. Пока устанавливается система, можете задать пользователя и пароль root — это намного интереснее, чем просто глазеть на строку состояния; но у вас все равно останется время на предписанную LXF

чашку чая. Начальная установка займет около 4 ГБ, по нынешним временам вполне стандартно.

Помощь разработчикам

Рабочий стол по умолчанию — Gnome 3.14, и он предлагает туманный фон в стиле Dreamworks в королевских голубых тонах, которые долгие годы являлись фирменным цветом Fedora. Хотя Gnome 3.15 появился в конце ноября, мы не рассчитываем, что его включат в этот релиз. Однако всегда есть возможности для обновления. Что касается приложений, то у нас имеются LibreOffice 4.3, Firefox 33, Shotwell, Rhythmbox, Transmission — вероятно, все они вам знакомы.

Возможно, менее знакомым окажется DevAssistant, проект Fedora, возникший года два назад. Цель DevAssistant — устранить из проектов программирования всякие сложности: настроить среду, установить зависимости и опубликовать код, чтобы программист мог сконцентрироваться на самом программировании, не отвлекаясь на вспомогательные задачи. Имеются здесь также

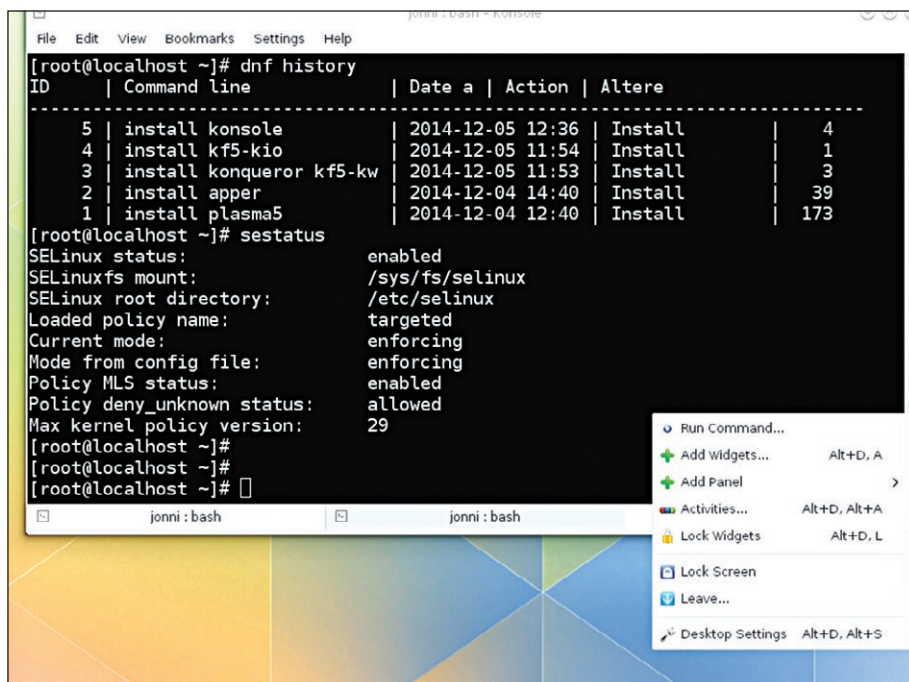
некоторые основные приложения Gnome, включая новое облачное приложение Documents (а также надежное Evince, его настольного двойника), приложение Maps (не то чтобы страшная угроза для Google), Cheese — сомнительного качества для Instagram/Snapchat, достойное приложение для записи диска Brasero и удобное приложение прогноза погоды. Кроме того, Gnome поддерживает потоковое воспроизведение мультимедиа DLNA, распределенный доступ к файлам через WebDAV и распределенное использование экрана через клиент Vino VNC.

Нет больше старого интерфейса PackageKit: его место занял ориентированный на приложения Software, который, без сомнения, более приемлем, чем Ubuntu Software Center.

Около 50% приложений Fedora идут с файлами AppData (предоставляющими скриншоты в XML, описания, переводы и рейтинги), требуемыми для отображения в приложении. Предоставлять файлы AppData Red Hat поощряет разработчиков вышележащего [upstream] ПО (вместо собственной команды, занимающейся пакетами) — недавно пересмотренный Gnome Software больше не является заповедником для пользователей Fedora, теперь поддержка имеется также для Arch, Debian и openSUSE. Далее, как уже упоминалось в статье «Управление пакетами: И как же оно работает?» [LXF186, стр. 50], AppData является частью движения в поддержку подхода всеохватывающего рабочего стола (который практикует Fedora) к приложениям типа Software Centre, так что со временем он станет удобен для использования в KDE и других рабочих столах.

Fedora также предлагает (не по умолчанию) сессию Gnome на Wayland, которая радует душу стабильностью. Очевидно, что на данный момент не все приложения поддерживают Wayland, но эти приложения элегантно вернулись к уровню совместимости XWayland. И хотя приятно слышать, что Wayland постепенно развивается, все же он (или, по крайней мере, та экосистема, которую он намерен поддерживать) пока что не используется в полной мере, так что это можно считать скорее предпросмотром, нежели чем-то еще.

У нас был отличный материал по управлению пакетами [снова LXF186, стр. 50], однако старый добрый Yum был там упомянут вскользь.



➤ Благодаря DNF установить 173 пакетов, составляющих Plasma 5.1, очень легко.

Спины и ремиксы

Хотя официальный релиз индивидуально настраивается по вашему усмотрению, вы, возможно, решите начать с одного из множества «спинов» Fedora. Есть релизы для совершенно определенных целей и задач. Так что вы можете попробовать KDE Spin (использующий Plasma 4 Desktop), или любой другой из спинов с разными рабочими столами, которые включают в себя, но не ограничиваются Xfce, LXDE и Mate-Compiz.

Помимо разновидностей рабочего стола, есть также Спины специально для ученых, для любителей роботов, для геймеров и для носителей

алюминиевых шляп (релиз, посвященный безопасности, с инструментами расследования, обнаружения вторжения и анализа сети).

Вы даже можете создавать собственные ремиксы, и если сочтете их стоящими, предложить их Fedora для официального утверждения, после чего они могут стать спином. Этот процесс хорошо задокументирован на https://fedoraproject.org/wiki/Spins_Process, но общая идея — начать с имеющейся готовой конфигурации kickstart из официального релиза или спина, и добавить (или убрать) то, что сочтете нужным. Приведа всё в соответствие

с основной политикой Fedora, следует предоставить свой результат на суд Spin Wrangler, который решает, является ли ваша Spin Page полной и соответствующей — и если да, то ваша работа будет отправлена дальше, в Spins SIG (Special Interest Group), где определят, достойна ли она представления Совету [Board] — это главный «хранитель печати». Со стороны может показаться, что здесь многовато политики и бюрократии, но нельзя же запросто позволить любому Тому, Рикерту или Гарриэт порочить высокое имя Fedora своим недопеченным программным продуктом.

Хотя он во многом сравним с *Apt* в Debian, есть у него несколько уникальных функций. *Yum* автоматически обновляет локальную базу данных имеющихся пакетов *SQLite*. Это означает, что вам не нужна отдельная команда **upgrade**, и вы никогда не столкнетесь с ситуацией, когда система пытается установить пакет старше того пакета, который имеется в репозиториях.

Еще одна замечательная функция — использование *delta-RPMs*, которое здорово экономит полосу пропускания скачиванием вместо полных пакетов лишь разницы [diffs], при наличии таковой.

Yum написан на Python, и его можно расширить через систему плагинов; функции *diff* создаются плагином под названием *Presto*, но он оказался настолько хорош, что в Fedora 19 его ввели в *Yum*. Официально одобренные плагины покажет команда

```
$ yum search yum-plugin
```

Вы увидите такие потрясающие плагины, как *fastestmirror* (который делает именно то, что указано на упаковке: ищет самое быстрое зеркало), *local* (который поддерживает локальный репозиторий всех установленных пакетов — очень удобно, если вы устанавливаете одни и те же пакеты на несколько компьютеров) и *merge-conf* (для отслеживания изменений в файлах настройки). Естественно, неофициальных плагинов еще больше — можете заняться их изучением.

Новый менеджер пакетов

Еще одна особенность *Yum* в том, что в Fedora 22 его заменит *DNF* (некоторые считают, что это сокращение означает *Dandified Yum*, однако официально заявлено, что оно вообще ничего не означает). Но волноваться не стоит: синтаксис остается практически таким же, и вы даже можете установить пакет *dnf-yum* в качестве оболочки *DNF*, чтобы он вызывался вместо *Yum*. *DNF* появился в Fedora 18, однако только с Fedora 20

он стал считаться готовым для повседневного применения. *DNF* — это ответвление *Yum*, с намного более аккуратной кодовой базой; этим мы частично обязаны заимствованию многих функций из внутренней библиотеки под названием *Hawkey*. Причины выбора другого менеджера пакетов (помимо практически неистребимого желания создавать ответвления) следующие:

» **Расширяемость** Хотя для *Yum* можно писать плагины, API не слишком хорошо задокументирован, и все должно делаться в Python. *DNF* (точнее, *Hawkey*) предлагает понятный API, доступ к которому можно получить через C и Python, и который, следовательно, намного лучше работает с такими интерфейсами, как *Gnome Software*.

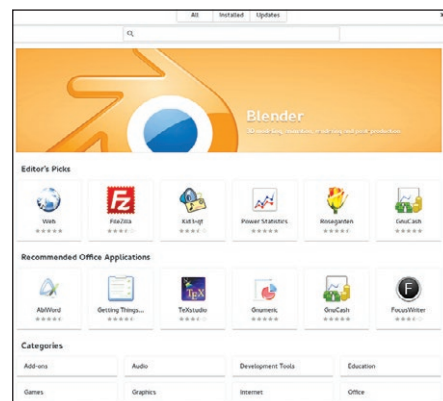
» **Разрешение зависимостей** Хотя постоянным пользователям это в принципе не заметно, *Yum* можно загнать в состояние неразрешимых зависимостей. *DNF* использует расширенную библиотеку *libsolv* openSUSE, которая отлично справляется с решением дилемм зависимостей.

» **Скорость** С точки зрения потребителя, *DNF* пошел на смелый шаг и синхронизировал метаданные в качестве фонового сервиса через процедуру *cron*, что означает следующее: сначала вы видите случайные всплески вашего интернет-трафика,

«Цель DevAssistant — устранить сложности из процесса программирования.»

и во-вторых, установка новых пакетов и обновление системы теперь задействует вдвое меньше строк состояния, чем раньше.

Помимо того, что это более симпатичный менеджер пакетов, *DNF* также упрощает использование новых репозиториях Copr. Они являются аналогами PPA в Ubuntu — помогают упростить процесс включения ваших программ



» **Gnome Software: новый дизайн как будто вытащил все самое броское из Ubuntu Software Center.**

в легко доступные репозитории пакетов. До Copr создание неофициальных пакетов включало использование Fedora Koji Build System или SUSE Open Build Service. И та, и другая система является всеобъемлющей, а будучи таковыми, обе весьма тяжеловесны.

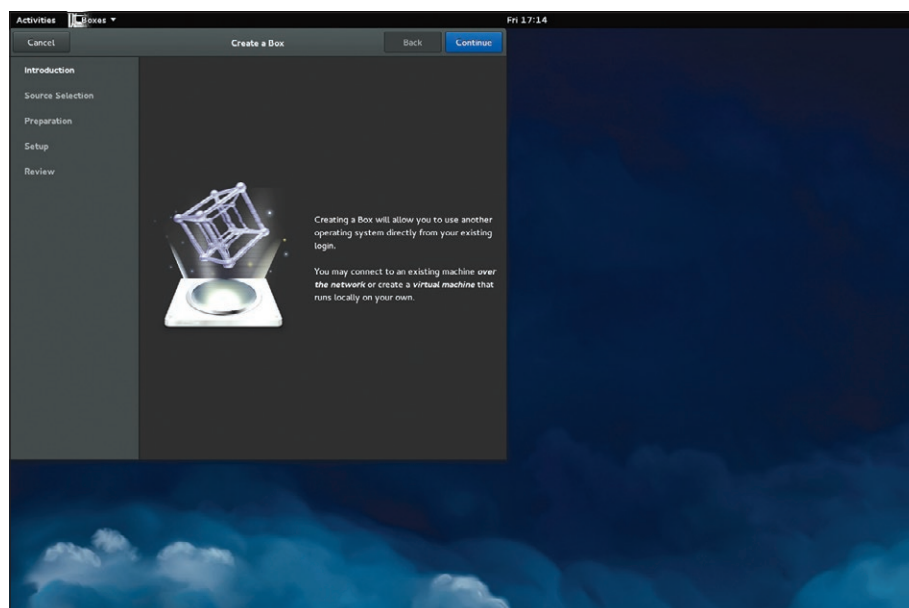
Более того, поскольку Koji — это также способ создания официальных пакетов Fedora (их не приносит аист, и в капусте их не находят), каждый, кто его использует, обязан выполнять драконовские требования и правила. Поэтому Copr определенно не замена Koji, во многом он является его противоположностью: это простая в использовании, легковесная и полнофункциональная система. Например, если вы хотите запустить Plasma 5.1, самый новый рабочий стол KDE, вы сначала включаете репозиторий Copr Даниэля Вратила [Daniel Vratil], и потом пусть *DNF* позаботится обо всем остальном:

```
$ sudo dnf copr enable dvratil/plasma-5
$ sudo dnf install plasma5
```

Вы можете установить пакет KDE 5 Application suite, добавив репозиторий *kde-applications-5*, как описано выше. Хотя внутренние библиотеки KDE 5 Frameworks уже имеются в стандартных репозиториях Fedora, Plasma и его приложения пока что считаются нестабильными и требуют встраивания, поскольку они конфликтуют со всей стабильной мелочевкой KDE 4.

Подводя итоги — Fedora 21 является отличным триумvirатом дистрибутивов, построенным на добротном наследии. Вновь провозглашенный тройственный подход является дерзким шагом, хотя мы считаем его необходимым, поскольку каждая редакция предназначена для определенной целевой аудитории: для Server не требуется среда рабочего стола, а рабочим станциям незачем запускать серверы LDAP.

Некоторые утверждают, что сама идея дистрибутива умирает, и что будущее за облачными технологиями и контейнерами — кто знает, что будет; но, как бы то ни было, эти три дистрибутива — по крайней мере, на данный момент — очень даже живые и энергичные. **LXF**



» **Boxes** облегчают настройку виртуальных машин. И мы еще заметили на фоне по умолчанию месяц и мальчика с удочкой, в стороне от всяких судебных исков.

Apt для Linux Mint

Подивившись отсутствию документации для этой полезнейшей утилиты, **Алексей Федорчук** решил восполнить пробел.

Мы рассмотрим особенности утилиты *apt* в реализации для дистрибутива Linux Mint и её отличия от семейства утилит из пакета *apt*, общего для всех Deb-дистрибутивов. В процессе написания обнаружилось, что для Linux Mint реализация утилиты *apt* не документирована никак — не только на языке родных осин, но даже на мове Вильяма нашего, Шекспира. И я решил посвятить ей отдельный очерк. Ведь многие начинающие пользователи Mint и особенно Linux вообще, судя по соответствующим сайтам, блогам и форумам, даже не подозревают о существовании реализации *apt* для Mint и её отличиях от тёмки из одноимённого пакета. И потому механически применяют рецепты для чистой Ubuntu и её прямых клонов, на которые так богаты указанные ресурсы. Хотя использование *apt* для Mint делает эти рецепты излишними — функциональность данной утилиты позволяет добиться той же цели быстрее и проще. По крайней мере, сократив количество нажатий на клавиши.

Обзор

Не путайте утилиту *apt* в реализации для Mint ни с одноимённым пакетом, входящим в состав всех Deb-дистрибутивов (в том числе и в Mint), ни с одноимённой же утилитой из этого пакета.

Утилита *apt* для Mint включена в состав пакета *mintsystem*, что она сама и покажет:

```
$ apt contains /usr/local/bin/apt
mintsystem: /usr/local/bin/apt
```

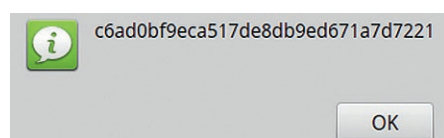
Стандартная утилита *apt* располагается в каталоге `/usr/bin`, а *apt* для Mint находится в каталоге `/usr/local/bin`, что демонстрирует такая команда:

```
$ which apt
/usr/local/bin/apt
```

При вводе в командной строке **apt** без указания пути вызывается именно она — это определено в переменной `PATH`, указанной в общесистемном конфигурационном файле `/etc/login.defs`:

```
$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin...
```

Так что для запуска стандартной утилиты *apt* из одноимённого пакета для неё следует указывать полный путь; например, для вывода списка установленных пакетов —



➤ Рис. 1. Вывод контрольной суммы командой `mint-md5sum`.

```
$ /usr/bin/apt list --installed
```

Это чуть ли не единственная функция стандартного инструментария пакета *apt*, отсутствующая в *apt* для Mint. Ибо последняя перекрывает почти все возможности утилит *apt-get* и *apt-cache*, большинство возможностей командного режима *aptitude*, а также выполняет некоторые функции низкоуровневой утилиты *dpkg*.

Пятеро из одного пакета

Утилита *apt* — не единственный компонент пакета *mintsystem*. Кроме неё, он включает ещё четыре утилиты (также помещённые в каталоге `/usr/local/bin`: *mint-md5sum*, *search*, *highlight* и *pastebin*). К управлению пакетами некоторое отношение имеет только первая из них, предназначенная для подсчёта контрольных сумм. Так, команда

```
$ mint-md5sum opera-stable_26.0.1656.60_amd64.deb
```

выведет её для пакета *opera-stable* в отдельном окне графического режима. (Рис. 1.)

Ищите — и обряцете

Утилита *search* из пакета *mintsystem* предназначена для поиска текстовых фрагментов в заданных каталогах. Формат её вызова таков:

```
$ search for [искомый фрагмент] in [каталог для поиска]
```

Например, в форме

```
$ search for 'дистрибутив Mint' in /home/current/alv.me
```

она отыщет все абзацы с вхождением 'дистрибутив Mint' во всех файлах указанного каталога.

Команда **search** не является полной заменой утилит семейства *grep*: в частности, она не поддерживает регулярные выражения. Но вполне может служить более простой в использовании альтернативой во многих тривиальных случаях, столь частых в практике применителей-текстовиков.

Удар в сетку

Утилита *pastebin* предназначена для быстрого размещения в Сети фрагментов текста, которые почему-либо нежелательно делать доступными каким-либо иным образом. Делается это через сервис, предоставляемый проектом Mint. Так, командная конструкция

```
$ echo 'Утилита pastebin предназначена для быстрого размещения в Сети' | pastebin
```

даст такой вывод:

```
http://paste.linuxmint.com/view/u5i0
```

То есть введённый фрагмент будет доступен (например, через браузер) по указанному в выводе адресу. Правда, русскоязычный текст по умолчанию окажется там в кодировке ISO 8859-5, так что

надо озаботиться тем, чтобы браузер поддерживал перекодирование страницы на лету.

Раскрась сам

Утилита *highlight* обеспечивает подсветку произвольного текстового фрагмента, заданного как её аргумент. Например, командная конструкция

```
$ echo 'Утилита <code>pastebin</code> предназначена...' | highlight code
```

выдаст подсвеченным фрагмент `code`. (Рис. 2.)

Теоретически рассуждая, если вывод этой конструкции передать по конвейеру команде *pastebin*, то и в Сети соответствующий фрагмент будет размещён в «подсвеченном» виде. Однако сервис проекта Mint такой возможности не поддерживает.

Применение

Утилита *apt* для Mint запускается одноимённой командой CLI с указанием внутренней субкоманды, определяющей цель действия и, в большинстве случаев, аргумента (аргументов), в качестве которых выступает имя пакетов (или имена — их может быть сколько угодно):

```
$ apt command pkgname1 ... pkgname#
```

Некоторые часто используемые внутренние команды *apt* аргументов не требуют.

Полный список внутренних команд *apt* для Mint можно получить «голой» командой

```
$ apt
```

вывод которой слишком длинен, чтобы его здесь приводить. Большинство важных для применения внутренних команд будет рассмотрено в соответствующих разделах этой статьи. А пока для начала следует сказать о внутренних командах **version** и **help**. Первая теоретически должна выводить номер текущей версии *apt* для Mint, но практически не выводит ничего — лишь возвращает подсказку командной строки. Команда же **help** без аргументов выведет список внутренних субкоманд, идентичный приведённому выше. При указании аргумента — любой из внутренних команд — она выведет её эквиваленты для *apt-cache*, *apt-get* или *dpkg*. Например:

```
$ apt help search
"apt search" is equivalent to "aptitude search"
$ apt help install
"apt install" is equivalent to "sudo apt-get install"
$ apt help deb
"apt deb" is equivalent to "sudo dpkg -i"
```

Внутренние команды *apt* для Mint можно разделить на четыре группы согласно их назначению, а именно — для

- 1 получения информации о пакетах;
- 2 установки и удаления отдельных двоичных пакетов;

```
[alv]$=> echo 'Утилита <code>pastebin</code> предназна
чена для быстрого размещения в Сети' | highlight code
Утилита <code>pastebin</code> предназначена для быстро
го размещения в Сети
[alv]$=> █ [~]
```

Рис. 2. Подсветка в выводе команды highlight.

- 3 общего обновления системы;
- 4 работы с пакетами исходных текстов.

Команды первой группы может выполнять обычный пользователь; второй, третьей и, обычно, четвёртой — только администратор. Но для получения таких прав утилите *apt* для Mint не нужна команда **sudo**, данная явным образом: **sudo** вызывается автоматически при попытке исполнения соответствующих внутренних команд. Например:

```
$ apt install geany
[sudo] password for alv:
```

Внутренние команды *apt* для Mint целесообразно рассматривать по указанным группам.

Информация о пакетах

Пакетный менеджмент начинается с поиска нужного пакета, для чего предназначена внутренняя команда **search**, требующая аргумента в виде ключевого слова. Поиск по ключевому слову осуществляется в именах пакетов и их кратких описаниях (т. н. резюме). Например, команда

```
$ apt search geany
```

отыщет одноимённый пакет для установки этого текстового редактора (называемого, однако, «Небольшой и быстрой IDE») и все его плагины. Важное отличие от аналога — команды **apt-cache search**: внутренняя команда **search** утилиты *apt* показывает основной и дополнительный статусы пакетов (см. врезку вверху стр. 59).

Поисковые задачи в отношении пакетов не ограничиваются отысканием их по имени или ключевому слову в описании. Так, нередко требуется отыскивать пакеты с тем или иным статусом, как основным, так и дополнительным. И *apt* для Mint даёт средства для этого — правда, в комбинации с другими утилитами.

В *apt* из комплекта *apt* есть очень полезная внутренняя команда **list**, которая с помощью опций **--installed** и **--upgradable** позволяет отсортировать установленные и доступные для обновления пакеты. В утилите *apt* штатного аналога её не имеется, однако тех же результатов можно достичь комбинацией её внутренней команды **search** с утилитой *grep*.

Для начала отмечу, что вывести список всех пакетов, имеющихся в подключённых репозиториях, можно таким образом:

```
$ apt search -
```

А дальше остаётся только воспользоваться тем, что вывод *apt* для Mint включает основной и дополнительный статус пакетов. То есть получить список установленных пакетов можно так:

```
$ apt search - | grep ^i
```

А автоматически установленных пакетов — так:

```
$ apt search - | grep ^i A
```

Этот подход пригоден для поиска пакетов с любыми другими основным и дополнительным

статусами, например, удалёнными без очистки от конфигов:

```
$ apt search | grep ^c
Можно отыскать в системе и «Сломанные» пакеты:
$=> apt search | grep ^i B'
```

В общем, возможности связки *apt* для Mint и *grep* ограничены только задачами применителя, его фантазией и знанием опций второй из этих утилит. Внутренняя команда **held** позволяет отсортировать пакеты с фиксированной версией, то есть те, что не обновятся по команде **apt upgrade**, о которой скоро пойдёт речь.

Подробную информацию об отдельном пакете можно получить с помощью внутренней команды **show**. В её выводе будут присутствовать:

- » имя пакета;
- » его основной статус (установлен или нет);
- » статус дополнительный (установлен ли автоматически);
- » номер версии и сборки;
- » приоритет (см. врезку);
- » раздел репозитория (admin, devel и т. д.);
- » майнтейнер и его контакты (например, Clement Lefebvre <clm@linuxmint.com>);
- » архитектура — i386 или amd64;
- » размер в распакованном виде;
- » зависимости, рекомендации, предложения и конфликты;
- » краткое описание, так называемое резюме;
- » URL сайта проекта.

Более подробные, чем вывод команды **show**, сведения о зависимостях пакета даёт пара внутренних команд **depends** и **rdepends**. Первая выводит полный список пакетов, от которых зависит заданный в качестве её аргумента — жёстких, рекомендуемых, предлагаемых и конфликтующих. Команда же **rdepends** решает обратную задачу: выводит список пакетов, зависящих от данного.

Все приведённые выше внутренние команды дают информацию как об установленных пакетах,

```

Пакет: geany
Состояние: установлен
Автоматически установлен: нет
Версия: 1.24.1+dfsg-1
Приоритет: необязательный
Раздел: devel
Сопровождающий: Geany Packaging Team <pkg-geany-team@lists.اليoth.debian.org>
Архитектура: amd64
Размер в распакованном виде: 3 072 k
Зависимости: libatk1.0-0 (>= 1.12.4), libc6 (>= 2.15), libcairo2 (>= 1.6.0), libfontconfig1 (>= 2.11), libfreetype6 (>= 2.2.1), libgcc1 (>= 1:4.1.1), libgdk-pixbuf2.0-0 (>= 2.22.0), libgl1.0-0 (>= 2.37.3), libgtk2.0-0 (>= 2.22.0), libpango-1.0-0 (>= 1.18.0), libpangocairo-1.0-0 (>= 1.14.0), libpangoft2-1.0-0 (>= 1.14.0), libstdc++6 (>= 4.1.1), geany-common (= 1.24.1+dfsg-1)
Предварительные зависимости: multiarch-support
Предлагаются: libvt9, doc-base
Конфликтуют: geany
Имяют: geany-plugins-common (< 0.21), geany-plugins-common (< 0.21)
Предоставляют: geany-abi-09, geany-api-217
Описание: fast and lightweight IDE
Geany is a small and lightweight integrated development environment. It was developed to provide a small and fast IDE, which has only a few dependencies from other packages. It is using only the GTK2 toolkit and therefore you need only the GTK2 runtime libraries to run Geany.

The basic features of Geany are:
* syntax highlighting
* code completion
* auto completion of constructs like if, for and while, XML and HTML
    
```

Рис. 3. Вывод утилиты apt для Mint.

так и о пакетах, доступных в подключённых репозиториях. А вот команды **contains** и **content** работают только для установленных пакетов. Первая позволяет определить, к какому пакету принадлежит данный файл — именно таким способом была определена выше принадлежность утилиты *apt*:

```
$ apt contains /usr/local/bin/apt
mintsytem: /usr/local/bin/apt
```

Внутренняя команда **content** выводит список всех файлов пакета с указанием их положения в файловой иерархии.

Сведения о смене версий пакета добываются с помощью внутренней команды **changelog**. А некоторые дополнительные сведения по этому вопросу даёт последняя из «информационных» команд — **policy**, требующая в качестве аргумента имя пакета, установленного или неустановленного. Она выводит номер его текущей версии (разумеется, только для установленного пакета) и версии — кандидата на установку, список версий, доступных в репозиториях, а также приоритет последних (см. врезку внизу стр. 58).

Работа с двоичными пакетами

Главное действие в отношении пакетов, которые были сочтены полезными — их установка. А основным инструментом установки является

Разные судьбы

Вокруг утилит под одним и тем же названием *apt* сложилась парадоксальная ситуация. Выход 1 апреля 2014 г. утилиты из одноимённого пакета, приуроченный к шестнадцатилетнему юбилею этого проекта, был отмечен прогрессивной общественностью и широко освещён на тематических ресурсах, хотя утилита эта весьма ограничена функционально и год спустя сохраняет статус

экспериментальной. В то же время реализация утилиты *apt* для Mint, существующая как минимум с 2009 года и доказавшая свою функциональность и устойчивость, практически не известна не только широкой публике, но даже применителям её «родного» дистрибутива. Вот уж действительно, Пророков нет в отечестве своём, Но и в других отечествах не густо...

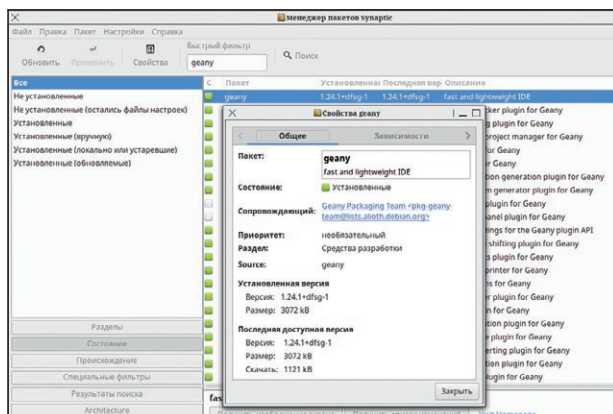


Рис. 4. Вывод через графический интерфейс Synaptic.

внутренняя команда **install**. В качестве аргументов она принимает имена пакетов — те самые, которые были найдены командой **apt search** и в полезности которых можно было убедиться командой **apt show**. Скажем, чрезвычайно полезный текстовый редактор *Geany* установит команда

```
$ apt install geany
```

которая сначала запросит пароль пользователя с административным типом учетной записи:

```
[sudo] password for alv:
```

а затем, считав локальный список пакетов и построив дерево зависимостей, сообщит о наличии таковых, объёме скачиваемых пакетов и занимаемом месте на диске после установки, и запросит подтверждение серьёзности намерений.

Согласие с установкой предполагается по умолчанию, так что тут достаточно нажать Enter. После чего начинается скачивание пакетов из содержащего их репозитория, распаковка и инкорпорация компонентов в файловую иерархию, а также регистрация в базе данных и включение, если требуется, исполняемого файла в главное меню (для *Geany* — в секцию Программирование, так как эта программа позиционируется её авторами как IDE — Integrated Development Environment, то есть интегрированная среда разработки). Основной статус пакета *geany* изменится на «установленный»:

```
$ apt search geany | head -n 1
```

```
i geany - Небольшая и быстрая IDE
```

А пакет *geany-common*, установленный как его зависимость, приобретёт ещё и статус автоматически установленного:

```
$ apt search geany-common
```

```
i A geany-common — Небольшая и быстрая IDE — общие файлы
```

Если в системе уже был установлен данный пакет более старой версии, он будет обновлён. А вот

переустановить пакет той же версии (например, если он был безнадежно испорчен в ходе экспериментов) команда **install** откажется, сообщив, что уже установлена самая новая версия *geany*.

Однако для таких случаев существует особая команда **reinstall**: её аргумент — установленный пакет, который нуждается в исправлении.

Локально отдельные пакеты можно установить внутренней командой **deb**, аргументом которой должно быть полное имя файла пакета

(если нужно, с указанием пути). Так, команда

```
$ apt deb sublime-text_3065_amd64.deb
```

установит текстовый редактор *Sublime*; разумеется, предварительно файл этого пакета должен быть скачан.

Поскольку внутренняя команда **deb** — полный эквивалент конструкции **sudo dpkg -i**, она не занимается разрешением зависимостей, а только сообщает об их нарушении. И, в отличие от внутренней команды **install**, команда **deb** не только обновит пакет до более новой версии, но и переустановит его версию текущую.

Установленные пакеты иногда требуется и удалять. Этой цели в *apt* для Mint служат две внутренние команды — **remove** и **purge**, аргументами которых служат, естественно, имена удаляемых пакетов. Первая удаляет файлы пакета, но сохраняет его общесистемные конфигурационные файлы, вторая — удаляет также и их. Различие между ними отражается в основном статусе удалённого пакета: в первом случае его значение будет **c**, во втором — **p**, как и у пакетов, которые никогда не устанавливались.

И **remove**, и **purge** автоматически удаляют все зависимые пакеты; список их выводится после ввода пользовательского пароля и перед запросом на подтверждение операции. Список этот нужно читать очень внимательно, чтобы случайно не удалить что-нибудь жизненно необходимое.

Пакеты, от которых зависит удаляемый, автоматически не удаляются ни **remove**, ни **purge**. В этом случае *apt* предлагает воспользоваться внутренней командой **autoremove** для очистки системы от «осиротелых» зависимостей:

```
$ apt autoremove
```

Она не нуждается в аргументах и выполняет свою работу молча, не задавая вопросов. Перед её

выполнением не вредно выполнить другую внутреннюю команду — **check**, проверяющую систему на предмет «сломаных» зависимостей. При хорошем раскладе после ввода пароля она некоторое время будет заниматься построением дерева зависимостей, а потом сообщит о завершении работы. При раскладе же плохом последует предложение воспользоваться опцией **--fix-broken** (или просто **-f**), о которой я расскажу несколько позже.

Удаление всех автоматически установленных пакетов не всегда желательно, даже если, по мнению системы, они кажутся «осиротелыми». Обычно это относится к пакетам, установленным в составе метапакетов или задач [tasks] — они часто требуют индивидуального обращения.

Избежать тотального удаления автоматически установленных пакетов в ходе выполнения операции **autoremove** можно, если предварительно снять с них дополнительный статус A. Для чего сначала нужно получить их описанным выше способом, выявить в нём кандидатов на лишение автоматического статуса, а потом осуществить это простым повторением команды установки:

```
$ apt install [pkgnames]
```

Перед установкой пакетов из репозитория они предварительно скачиваются и помещаются в каталог **/var/cache/apt/archives/**. Со временем файлов пакетов накапливается много, а нужны они бывают только в исключительных случаях. Для избавления от них в *apt* для Mint предусмотрены команды **autoclean** и **clean**. Первая удаляет из кэша только пакеты устаревших версий, сохраняя те, версии которых установлены в системе. Вторая же полностью очищает указанный каталог.

Обновление системы

Сказанное выше касалось единичных пакетов или их серий — любая из перечисленных субкоманд принимает любое количество аргументов. Однако в утилите *apt* предусмотрены и внутренние команды для общего обновления всех установленных пакетов, а также для тотального — системы. Однако, прежде чем выполнить любую из них, необходимо провести обновление локального кэша пакетов, то есть получить списки новых и обновлённых пакетов. Делается это внутренней командой **update**, не нуждающейся в аргументах:

```
$ apt update
```

Её же обязательно следует выполнять после каждого изменения в репозиториях — подключения новых или отключения имевшихся. Теоретически для редактирования списков репозиториев в *apt* для Mint предназначена команда **sources**. Однако практически она бесполезна, так как вызывает текстовый редактор по умолчанию (*nano*) для редактирования **/etc/apt/sources.list**. В нашем же дистрибутиве этот файл содержит только репозиторий локального оптического диска, а все реально подключённые репозитории описываются в файлах каталога **/etc/apt/sources.list.d**.

Для обновления всех, по возможности, пакетов установленной системы в *apt* для Mint существует внутренняя команда **upgrade**. Она выявит все пакеты, для которых в репозиториях доступны более свежие версии, выведет их список, объём

Приоритеты пакетов

Приоритет пакета устанавливает его важность для функционирования системы. По этому признаку выделяются пакеты обязательные [required], без которых система не может нормально работать; важные [important], существенно влияющие на её функциональность; стандартные [standard],

обычно присутствующие в любой Linux-системе; дополнительные [optional], теоретически не являющиеся обязательными, но практически необходимые применительно; и «экстраординарные [extra]», которые могут конфликтовать с пакетами всех остальных приоритетов.

для скачивания и прирост объёма занятого дискового пространства после выполнения процедуры, а также запросит подтверждение.

В ходе выполнения **upgrade** обновляются по возможности все пакеты, за исключением тех, для разрешения зависимостей которых обновление потребует доустановки новых пакетов или удаления существующих. Для таких пакетов текущие версии будут сохранены.

При использовании команды **upgrade** следует учитывать, что **upgrade** обновляет в том числе и те компоненты, которые по умолчанию заблокированы для обновления через фирменный инструмент **mintupdate** (о котором говорилось в статье про LMDE из **LXF194**) — ядро и всё, что с ним связано, **glibc**, и так далее (пакеты уровней 4 и 5). Так что, прежде чем применять эту команду, следует либо всё взвесить и решиться на обновление указанных компонентов, либо явным образом зафиксировать их версии.

Фиксация версий пакетов может потребоваться и в ряде других случаев — например, при использовании более не поддерживаемых, но по-прежнему необходимых пакетов, пакетов, пересобранных с собственными опциями, и ещё некоторых. Она выполняется внутренней командой **hold** с указанием имени фиксируемого пакета (пакетов). После чего пакет приобретает основной статус **h** и не затрагивается обновлениями. Обратная процедура, то есть снятие фиксации, если в ней пропала необходимость, выполняется внутренней командой **unhold**.

Для тотального обновления системы предназначена внутренняя команда **dist-upgrade**: она не только обновляет все пакеты, для которых обновления доступны, но может доустанавливать новые пакеты и удалять существующие, если это необходимо для разрешения зависимостей. Эта субкоманда применяется, например, при смене релиза дистрибутива — например, в статье из прошлого номера рассматривался переход с LMDE 1 на LMDE 2 Betsy. Для чего было достаточно прописать соответствующие репозитории в файлах их описаний, а затем дать команду

```
$ apt dist-upgrade
```

Есть и ещё несколько случаев, требующих применения **dist-upgrade**, а не просто **upgrade** — например, обновление версии рабочей среды и некоторых других базовых компонентов системы.

При тотальном обновлении через **dist-upgrade** следует помнить о том, что выше было сказано про **upgrade**. И если в данном случае обновление базовых компонентов системы (ядра и так далее) необходимо, то о фиксации самосборных

Статусы пакетов

В Deb-дистрибутивах каждый пакет обязан иметь основной статус, и может (но не обязан) иметь статус дополнительный. Основные статусы таковы:

- » **i** — установленный пакет;
- » **p** — пакет не установленный или удаленный «вчистую» (т.е. вместе с файлами конфигурации);
- » **s** — пакет, деинсталлированный с сохранением конфигурационных файлов;
- » **v** — виртуальный пакет, объединяющий группу обычных пакетов, обеспечивающих аналогичную функциональность.

А дополнительные (необязательные) статусы бывают следующие:

- » **A** — установленный автоматически, как зависимость другого пакета; пакеты, не имеющие статуса **A**, считаются установленными вручную;
- » **h** — пакет с фиксированной версией (то есть не подверженный апгрейду);
- » **u** — пакет распакованный, но не установленный;
- » **H** — «недоустановленный» пакет;
- » **C** — пакет установленный, но не настроенный;
- » **B** — «сломанный» пакет, то есть установленный с нарушением зависимостей.

В сущности, все действия по управлению пакетами в Mint сводятся к изменению их статуса.

и неподдерживаемых пакетов посредством команды **hold** нужно позаботиться заблаговременно.

Команда **dist-upgrade** не всегда выполняется гладко и может завершиться сообщением об ошибке, связанной с нарушением зависимостей.

Работа с пакетами исходных текстов

Всё сказанное выше относилось к двоичным пакетам. Однако в утилите **apt** предусмотрены также и средства для работы с пакетами исходных текстов. Так, с помощью внутренней команды **source** можно просто скачать пакет, указанный в качестве её аргумента — разумеется, для этого должен быть подключён репозиторий исходников. Внутренняя команда **build** (эквивалент **sudo dpkg-buildpackage**) выполнит построение двоичного пакета (что требует соответствующего инструментария в установленном виде). А внутренняя команда **build-dep** ограничится конфигурированием необходимых для этого зависимостей. Например, команда

```
apt build-dep ubuntu-zfs
```

потребуется мне при сборке пакетов поддержки ZFS on Linux.

Итог

Можно видеть, что и по части манипулирования пакетами возможности утилиты **apt** широки и многогранны. То есть это действительно универсальное средство управления пакетами, в обыденной жизни способное почти всегда заменить все прочие — от низкоуровневой **dpkg** (обращение к которой потребуется только в исключительных случаях) до графического интерфейса **Synaptic**. Ибо не уступает последнему в наглядности вывода



» Рис. 5. Вывод Менеджера программ.

информации о пакетах, позволяя манипулировать ими проще и быстрее. Что же касается Менеджера программ, о котором шла речь в прошлой статье, то он выигрывает у **apt** только в том, что показывает картинку. Информативность вывода этих трех утилит можно сравнить по рис. 3—5.

Рядом с **apt** для Mint его тезка из одноимённого пакета (общего для всех Deb-дистрибутивов) Debian/Ubuntu выглядит ограниченным функционально, а традиционные **apt-cache** и **apt-get** — несколько усложнёнными синтаксически. Что же до **aptitude**, то она в этом контексте кажется вообще излишней: **apt** для Mint обеспечивает почти все функции её командного режима, а в интерактивном режиме эта программа в дистрибутивах семейства Ubuntu и её клонов уже давно работает не вполне корректно. **LXF**

Управление приоритетами

В репозиториях дистрибутива разных версий присутствуют, очевидно, разные версии пакетов. По умолчанию пакет устанавливается или обновляется из репозитория имеющейся на данной машине версии системы. Однако это можно изменить, задав для некоего пакета приоритет репозитория более

старой или более новой версии. Приоритет репозитория, в который входит пакет — это число; оно берётся из файлов каталога **/etc/apt/preferences.d**. Больше число — больше и приоритет.

Для управления приоритетами в утилите **apt-cache** дистрибутива Debian была придумана

внутренняя команда **policy**. Она облегчает совмещение в одной системе пакетов из многочисленных веток этого дистрибутива — Stable, Testing, Unstable, Experimental. В дистрибутиве Mint применение её ограничено, так как его тестовый репозиторий, так называемый **hometo**, по умолчанию не используется.

Новый web-браузер Vivaldi

Петр Семилетов, не смирившись с реформами браузера Opera, нашел продолжателя былых традиций.

Вообще я пользуюсь несколькими браузерами. С давних пор это *Firefox* и *Opera*. *Firefox* у меня основной: там есть нужные мне плагины, без которых как без рук. Например, *Scarbook* — безразмерный сарай для сбрасывания в него «на потом» страниц, которые хочется оставить. Или *AdBlock Plus*, с которым страницы так же чисты от рекламы, как улицы в Советском Союзе.

Прижился у меня и *Chromium*. Ибо когда я запускаю Карты Google в режиме «Просмотр улиц», мне становится очевидным, в каком браузере они работают лучше всего.

А к *Opera* я привык еще с начала века, когда сидел под Windows, а в Интернет выходил через телефонный модем на 36 килобит. Более того, жива еще была любительская сеть FidoNet, и свежие версии *Opera* мне было удобнее скачивать через нее.

Нельзя сказать, чтобы *Opera* с тех пор не менялась. Порой в худшую сторону — например, в верхней части окна маячила здоровенная рекламная область. Однако *Opera* была самой быстрой. Самой передовой. В ней впервые появился интерфейс на основе вкладок. При медленной скорости доступа к Сети, часто приходилось отключать картинки — в *Opera* это осуществлялось на лету, нажатием одной кнопки. Полноценный менеджер скачиваний файлов, до которого поныне многие браузеры почему-то не созрели. Уйма настроек.

Opera для Linux стала исправно выходить с пятой версии, в 2001 году. Да, закрытый исходник. Да, я тоже предпочитаю Свободное ПО, но удобство и стабильность *Opera* были мне важнее.

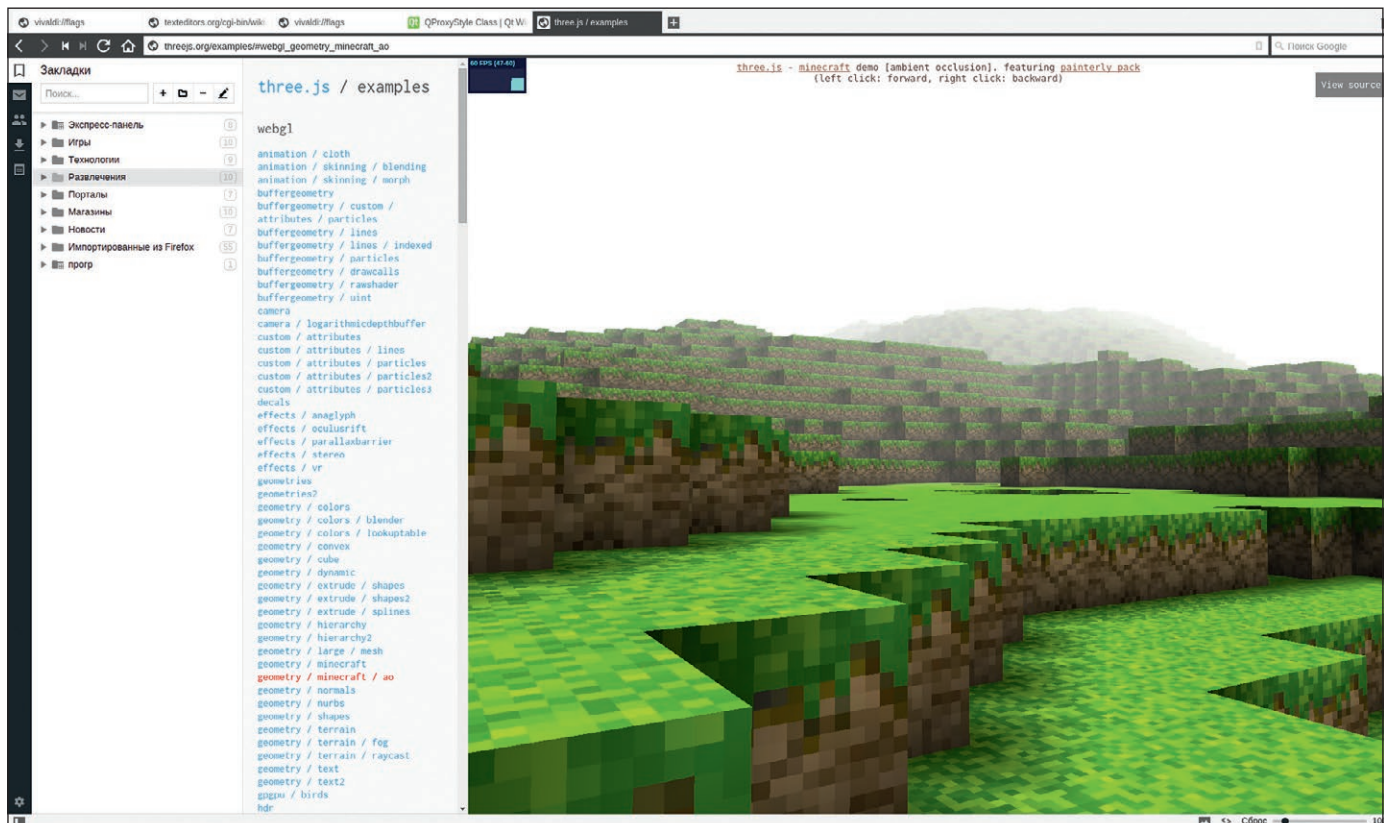
Затем, к 2013 году, появились тревожные звонки. Мол, движок *Opera* — Presto — устаревает!

Всё труднее поддерживать его в согласии с новыми технологиями! И было решено использовать движок от Google — сначала WebKit, потом его развитие — Blink. Тот же, что в *Chromium*. На движке Presto осталась только *Opera Mini*, а обычная *Opera* да *Opera Mobile* перешли на Blink.

Разработчики заявили, что теперь-то смогут сосредоточиться на самом браузере, не движке. Сделают всё еще лучше и удобнее. Многие пользователи, когда узнали, что им готовят взамен старой доброй *Opera*, принялись возмущаться. А другие, наоборот, радовались.

Летом 2013 года вышла последняя версия *Opera* для Linux на прежнем движке, *Opera 12.16*. На новом движке Linux-версии стали делать только через год, когда счетчик достиг 26.

А с 2001 по 2014 годы существовал сайт-сообщество MyOpera, нечто вроде социальной сети, образовавшейся вокруг популярного браузера.



➤ *Minecraft* в браузере Vivaldi! Воксельная графика, обеспеченная движком three.js.

В 2013 году в *Opera* надумали закрыть сообщество в 2014-м. Причина — поддерживать работу сообщества трудоемко, да еще и конкуренты больно сильны.

В конце 2013 года случилось событие, прошедшее мимо тех, кто не участвовал в *MyOpera*. Один из основателей *Opera Software*, бывший исполнительный директор компании, исландец Йон Стефенсон [Jon Stephenson] основал новую компанию — *Vivaldi Technologies*, и запустил сайт Vivaldi.net. В «музыкальности» названия ясно прослеживалась преемственность от *Opera*.

Vivaldi.net предоставил всё то, что имелося на *MyOpera* — почту, форум, чат, блог и возможность делиться фотографиями. Любители *MyOpera* перешли туда. Таким образом возникла начальная пользовательская база для того, о чем остальной мир узнал в январе 2015 года.

А в январе выложили «техническую предварительную версию» нового браузера — *Vivaldi*. Чтоб посмотреть и оценить задумку. Ошибки? Так это ж техническая предварительная версия!

Однако я скачал и попробовал. Зачем мне четвертый браузер? «Старая» *Opera* больше не обновляется, застряла на 12-м выпуске. Новая *Opera* воспринимается мною как совсем другая программа, у которой от прежней только имя. Я хотел бы найти замену именно старой *Opera*, ведь рано или поздно она, будучи продуктом с закрытым кодом, перестанет запускаться в моей системе. Поэтому с *Vivaldi* я связал надежду возрождения традиций прежней *Opera*. Насколько это удалось разработчикам?

Покамест судить трудно. Но начало хорошее.

Скачанный мною 64-битный RPM-пакет отлично установился в моей Mageia 4. К слову, доступны 64 и 32-битные пакеты как RPM, так и Deb. Не одному Linux такой почет — выложены также версии для Windows и Mac OS. Но мы-то знаем, за какой системой будущее.

Еще доступны для скачивания (ссылки даются в сообществе *Vivaldi* и связанных с проектом соцсетях) сырые выпуски, «снэпшоты», где происходит обкатка новшеств. Обкатывают, понятное дело, сами пользователи, а кто не ленится, сообщает разработчикам об ошибках. Русский перевод в «снэпшотах» неполный. Вот бы, кстати, обозначить русским словом это «snapshot». А то прижилось уже, да нет разве в нашем языке замены? Может — снимок [в *LXF* употребляется «моментальный снимок», — прим. ред.], срез. Но как внедрить в сознание, чтобы пользовались?

Итак, запускаю *Vivaldi*. Ожидал увидеть привычную, классическую *Opera*. Не она. Однако её развитие — определёнno.

Обо всём по порядку, но сначала про движок. Увы, не Presto. Взяли всё тот же Blink. Теперь, пожалуй, из крупных браузеров только у *Firefox* свой движок (Gecko), а так все работают на Blink или на WebKit. Blink лицензирован под LGPL и BSD.

О воспроизводимых форматах. Сразу отмечу: сказанное справедливо на время написания статьи. Конечно, всех интересует видео. В частности, на Youtube и ВКонтакте. При заходе на Youtube *Vivaldi* показывает видео через плеер



➤ Интерфейс *Vivaldi* — простой и ясный.

на технологии HTML5. При этом не поддерживается воспроизведение видео, сжатое кодеком H.264. Это значит, что Youtube дает посмотреть только видео в формате WebM VP8.

При этом получается странная картина! Тот же HTML5-плеер в Chromium, на том же видео, позволяет смотреть в HD-разрешении. А *Vivaldi* может предложить выбор HD-разрешений, а может и не предложить, урезав до 360p. Таких видео, что в разных браузерах доступны под разными разрешениями, я встречал несколько и не знаю, чем это обусловлено.

Видео ВКонтакте завязано на Flash. Вообще много чего завязано на Flash. А по умолчанию,

«Новшества обкатывают, понятное дело, сами пользователи.»

Vivaldi не подхватывает стандартный Flash-плагин от Adobe. Что же делать?

В состав Google *Chrome* (не *Chromium*) входит альтернативный плагин Flash — *Pepper Flash*. Но если просто установить Google *Chrome*, *Vivaldi* всё равно этот *Pepper Flash* не увидит.

Давайте проверим, где *Vivaldi* его ищет? Два способа. Первый — посмотреть исходный файл `/usr/bin/vivaldi` (или `vivaldi-snapshot`). Второй — в адресной строке браузера дать команду `vivaldi://about`. В появившейся странице, кроме прочего, будет выведена строка запуска *Vivaldi* и его параметры. Среди них вы и увидите ответ на вопрос:

```
--ppapi-flash-path=/usr/lib/PepperFlash/libpepflashplayer.so
```

Итак, *Vivaldi* ищет плагин в каталоге `/usr/lib/PepperFlash/`. Отлично! Скачиваем Google *Chrome*, открываем его дистрибутив как архив (с этим отлично справляется *Krusader* посредством сервиса

krarc — отдельным архиватором *Ark* либо через любой другой архиватор с поддержкой распаковки пакетов). Отыскиваем в нем каталог *PepperFlash* — он находится в `/opt/google/chrome`. Под правами root, копируем весь каталог в `/usr/lib`. Либо, установив *Chrome*, исправляем пути в файле запуска *Vivaldi*. Как вам удобнее.

Перезапускаем *Vivaldi*. Проверяем через адресную строку, видно ли *Pepper Flash* на странице плагинов:

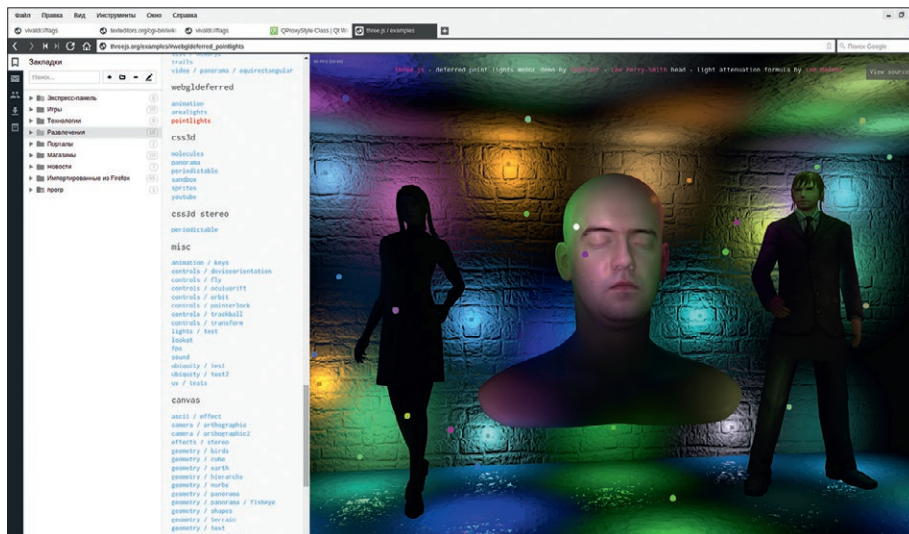
```
vivaldi://plugins
```

Должно быть два плагина — Adobe Flash Player (это на деле и есть *Pepper Flash*) да Chrome PDF Viewer (средство просмотра PDF). Заходим на любой сайт, где необходим Flash, тот же ВКонтакте или игровые ресурсы... Ура, видео и Flash-игры работают!

Как насчет другой, набирающей силу технологии — HTML5-игр с графикой WebGL? Браузер держит ее хорошо, если драйверы видеокарты позволяют. Для проверки достаточно зайти на страничку интерактивных примеров одного из лучших HTML5-движков — *three.js* (threejs.org/examples). Там же заодно можно сразу увидеть исходный код.

Вообще в Сети сейчас полно HTML5-игр, как и портов на HTML5 старых игр. Есть даже эмулятор *jsDosBox* для запуска игрушек от DOS. Попробуйте поиграть через него в классический *Doom* (jsdosbox.appspot.com) — скорость не такая замечательная, как в «родных» HTML5-играх, но всё равно здорово! Что можно выжать из HTML5, очевидно на демонстрационном разделе сайта Mozilla (developer.mozilla.org/ru/demos). Там выложена полноценная 3D-стрелялка *Bananabread*.

Но это передний край технологии. Большинство пользователей предпочитает другие игры, требующие меньше ресурсов — хотя, благодаря выдающимся способностям некоторых



➤ Весь арсенал современной 3D-графики прямо в окне браузера.

программистов, простенькие цацки порой оказываются прожорливей иных тяжеловесов.

Если для чтения новостей, сидения в соцсетях и просмотра онлайн-видео возможностей старой *Opera* еще вполне хватает, то именно для игр *Vivaldi* подходит больше. Я намеренно не сравниваю *Vivaldi* с «новой» *Opera*, поскольку пользователи *Vivaldi* — это в основном пользователи старой *Opera*, версии 12.

Чем же порадует их *Vivaldi*?

Запускаю его и вижу знаменитое окно Speed Dial, оно же «Экспресс-панель». Причем удобнее, нежели в прежней *Opera*. Можно создавать новые папки — экраны со ссылками на сайты. Такое в *Firefox* возможно при помощи плагина FVD Speed Dial. Но *Vivaldi* идет дальше: поддерживаются вложенные папки. Это настоящий рай для любителей всё упорядочивать.

На Экспресс-панели доступны также кнопки Закладки да История, открывающие весьма удобные тематические страницы. К Закладкам можно добраться и через левую панель, но там несколько сокращенный интерфейс.

Нам стоит остановиться на этих Закладках поподробнее.

Изначально, для них уже созданы папки-рубрики — Игры, Новости и тому подобное. Через контекстное меню любую рубрику можно добавить на Экспресс-панель, в качестве нового экрана. Мне это очень понравилось. Папки, созданные в Экспресс-панели, становятся папками в Закладках. Но если вы убираете папку с Экспресс-панели, она остается нетронутой в Закладках.

А вот отдельный сайт из Закладок на Экспресс-панель поместить нельзя. Зато каждой закладке можно дать описание и короткое имя. Вводя в адресную строку короткое имя вместо адреса, мы быстро попадаем на связанный с коротким именем сайт.

Обнаружив на странице Закладок кнопку Импорт, я нажал ее, и мне предложили импортировать в *Vivaldi* закладки, пароли и замет-

ки из других браузеров — *Opera*, *Firefox*, либо из HTML-файла с закладками. Я выбрал *Firefox*. *Vivaldi* довольно ошутимо задумался, я уж думал — подвис, однако нет. Долго ли, коротко — в Закладках появилась вся моя структура закладок из *Firefox*, а также при заходе на сайты стали подсказываться логины и пароли.

Левая панель *Vivaldi* сходна с *Opera*, но еще не все задумки воплощены в жизнь. Собираются сделать почтовый клиент. Как вы помните, раньше в *Opera* был встроенный почтовик. Теперь он отделен в проект Opera Mail. А в *Vivaldi* хотят вернуть отнятую у *Opera* функциональность. Под это дело уже готова еще одна вкладка левой панели — Контакты. В ней можно создать контакт и вызвать для него пункт контекстного меню Compose [Составить]. При этом открывается пустая вкладка. Надо полагать, в будущем в ней будет окно для составления письма. Если для Контакты не предусмотрено какое-либо взаимо-

«Стоит остановиться на этих Закладках поподробнее.»

действие с социальными сетями или видеозвонками, не совсем понятно, зачем выносить Контакты отдельно? Им тогда место в почтовом клиенте.

Далее, вкладочка Загрузок. До менеджера загрузок из *Opera* еще очень далеко: это простейшая качалка, сохраняющая файл в каталог **Загрузки**, причем нельзя даже выбрать другую папку. Что же, подождем.

И последняя вкладка — Заметки. Я и в *Opera* не понимал, зачем они нужны. Web-страницу в Заметки не сохранишь. Писать примечания для ссылок? Это уже есть в закладках. Вести какие-то свои заметки? Для сего существуют более удобные программы.

Отображение левой панели переключается кнопкой в левом нижнем углу окна. Панель можно растягивать до нужной ширины.

Есть еще нижняя панель. На ней располагаются ползунок масштабирования, кнопка сокрытия картинок и довольно странный инструмент под названием Эффекты страницы. Это список, где можно выбрать графические фильтры для всей страницы либо одних картинок — скажем, сделать их черно-белыми или с наложением эффекта сепии. Перевернуть цвета страницы. Включить принудительно моноширинный шрифт. Блокировать некий «контент» (так и написано, Content blocker) — исчезают некоторые Flash-баннеры.

Интерфейс *Vivaldi* насыщен не вполне очевидными, но удобными функциями. Например, в контекстном меню для корешка вкладки скрывается пункт меню Создать закладки всех открытых страниц. И поди догадайся, что там такое есть! Или, если удерживать левую кнопку мыши нажатой над кнопкой Назад (на панели инструментов), отобразится история переходов в текущей вкладке.

Со вкладками можно делать что угодно — клонировать, открывать в новых окнах, закрывать все другие, кроме текущей, и так далее. Размещение панели с корешками настраивается далеко. Надо пойти в Настройки > Внешний вид > Вкладки. Там же, во Внешнем виде (будто есть еще внутренний!), задается положение адресной строки и прочих частей интерфейса. Адресная строка, кстати, имеет неудобную привычку подсовывать «подсказку» из закладок вместо набираемого адреса, как бы вы ни хотели перейти именно по введенному адресу. Это нуждается в доработке!

Кнопки на панель рядом с адресной строкой добавлять нельзя. Жесткий набор: перемотка, перезагрузка, домашняя страница. Размер кнопок менять тоже нельзя. Они больше, чем в *Chromium*, и меньше, чем в *Firefox*. Пользователей с плохим зрением это вряд ли воодушевит, разве что цвета хорошо заметны: белый на красном. Они, однако, могут раздражать. Но выбор цветов или тем оформления пока отсутствует, хотя это и привлекло бы к сообществу вокруг браузера большее количество людей.

Настройки *Vivaldi* вынесены в отдельное окно, которое зачем-то можно открывать в несколько копий. Это неряшливость программирования. Кстати, окна *Vivaldi* лишены рамок, и когда белое окно Настроек появляется поверх белого же фона главного окна, впечатление неряшливости усиливается. Еще одна раздражающая мелочь — главное окно не сворачивается по нажатию на кнопку Свернуть. Оно как Ванька-встанька. Нажимаю кнопку Свернуть — свернулось, а глядишь, миг спустя снова развернулось. Выход из положения — щелкать по *Vivaldi* на панели задач. Сворачивает — разворачивает.

Но вернусь к Настройкам. Про раздел Внешний вид я уже говорил. Есть еще Основные, про них и рассказывать нечего, всего-то параметры домашней страницы да назначение *Vivaldi* основным браузером. Пока рановато! Доведите до ума, а потом делайте такую кнопку — с удовольствием нажму. У меня сейчас системный браузер — старая *Opera*.

Раздел Управление с иконкой, изображающей компас. Тут можно добавить новые поисковые движки, а также определить сочетания клавиш. Лично я бы перенес Поиск в Основные или куда-нибудь еще.

Раздел Безопасность. Настройки куков, менеджер паролей. Пароли не отображаются, зато видны логины. Хотя бы уже сделали опцию показа паролей, а то многие разработчики браузеров держат пользователей за воров собственных паролей. Я понимаю, в каком-нибудь интернет-кафе нужно пароли скрывать, но дома — дайте человеку опцию отображения паролей. Это же легко. Вот Firefox так умеет: Настройки > Защита > Сохраненные пароли > кнопка Показать пароли. Конечно, вы об этом не знали!

В Vivaldi, в Безопасности, присутствует целый раздел опций, являющихся наследием Google. Они так и называются — Google services.

И некоторые по умолчанию включены! Это Enable Google phishing and malware protection [Активировать защиту Google от вредоносного ПО] и Use Google to resolve navigation errors [Использовать Google для разрешения ошибок навигации]. В Firefox такое тоже предусмотрено (Настройки > Безопасность > Блокировать сайты, подозреваемые в атаках/мошенничестве). Технология подразумевает отправку «партнерам» данных о посещаемых вами сайтах.

Наконец, лишний раздел окна Настроек — Показать всё — отображает параметры всех перечисленных выше страниц.

Может, я чрезмерно строг к браузеру, который доступен лишь как «техническая версия», однако умалчивать о недоработках не хочется. В каждой программе есть плохое и хорошее. Когда о плохом говоришь, оно может исчезнуть: его исправят. Вот, например, в Vivaldi торчат уши из браузера

Chromium, вплоть до некоторых подписей; Впрочем, разработчики не скрывают, что в основе Vivaldi, как и в основе Google Chrome, лежит Chromium. И уйма сторонних библиотек, полный список коих доступен, если в адресной строке дать команду `vivaldi://credits`.

Более того — еще совсем недавно Vivaldi, после незначительных танцев с бубном, поддерживал некоторые расширения от Chromium/Chrome и Opera. А теперь перестал. Свои расширения, как обещают разработчики, будут. И я понимаю, зачем отключили сторонние: причина — надежность и независимость. Ибо если поддерживать расширения другого браузера, постоянно придется подстраиваться под этот другой браузер, под его возможности. Но учитывая общую основу исходников Vivaldi с Chromium и современной Opera, полагаю, что разработчикам расширений не состави

«Заменит ли Vivaldi „старую“ Opera? Да, и очень скоро.»

трудно выпускать версии для Vivaldi, как только поддержка в нем расширений будет объявлена официально.

Что покамест отличает Vivaldi от других браузеров? Как он воспринимается?

Как сырая версия следующего поколения «старой» Opera, лишенной множества привычных функций. Полезность новых функций, которых не было в Opera, для меня не очевидна. Так, по F2 вызывается средство Быстрые команды, где в отдельном меню доступны открытые вкладки и часть пунктов из главного меню. Для чего это сделано? Приятно, конечно, что те возможности, что были под рукой в Opera, здесь тоже под ру-

кой — ползунок масштабирования (да и клавиши плюс/минус работают), или корзина — в ней прячется список закрытых вкладок. Кстати, последнюю закрытую вкладку можно быстро открыть заново, нажав Ctrl+Z. Удобно, если вы захлопнули её по случайности.

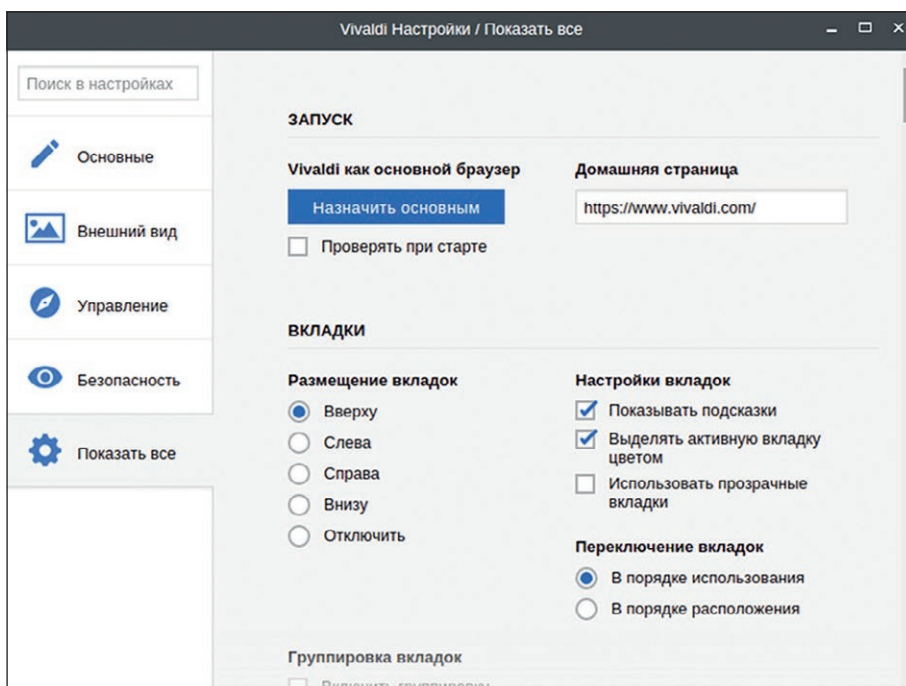
В интервью с командой Vivaldi не раз подчеркивается, что это браузер для «гиков», продвинутых пользователей. Надеюсь, он станет таким. Впрочем, старая Opera была продуктом для всех категорий пользователей; просто те, кто хотел от браузера большего количества функций и настроек, получали их без труда. А сейчас Vivaldi в самом деле браузер для продвинутых. Хочешь поддержку Flash? Изволь пошаманить!

Будущее. Проект живет и развивается. На vivaldi.net обсуждаются новые версии, разработчики проводят голосования по очередности внедрения новшеств. Пользователи могут предлагать что-то свое и участвовать в переводе интерфейса — жаль, что не в работе над кодом. Код закрыт. Насколько быстро функции, за которые пользователи голосовали, будут воплощены в новых версиях — посмотрим. Как программист, я вижу, что некоторые задачи решаются за пять минут (например, двойной щелчок мыши для закрытия вкладки). Движок управления сессиями можно написать за час. Вот поддержка тем оформления требует большего количества времени — и оно нужно не на само программирование, а на продумывание архитектуры.

Некоторые вещи из запланированных уже доступны. Как это? А очень просто. Наберите в адресной строке: `vivaldi://flags`.

Ого! Недокументированные функции! Здесь их можно включать и выключать. Плавная прокрутка? Пожалуйста. Сохранение web-страниц в формат MHTML, когда всё помещается в один файл — тоже здесь спрятано. Из полезного также — Включить функцию возобновления скачивания. Можно еще поиграться с настройками отрисовки, повысить производительность и так далее — на свой страх и риск. Страница Flags оснащена предупреждающим знаком радиационной опасности. Кстати, почти все опции на ней переведены на русский и снабжены относительно подробным описанием. Эту страницу можно смело поместить в окно Настроек, назвав раздел с ними как-нибудь вроде Дополнительные или Экспериментальные.

Что же, подыбем итоги. Можно ли пользоваться Vivaldi каждодневно уже сейчас? Да, можно. Лично для меня — даже в текущем состоянии Vivaldi удобнее, чем Chromium, и решающую роль здесь играют, как ни странно, всего две вещи: ползунок масштабирования и удобнейшая Экспресс-панель. Заменит ли Vivaldi «старую» Opera? Да, и очень скоро. Насколько хорош Vivaldi по сравнению с другими браузерами? Покамест их нельзя сравнивать по причине ранней стадии развития самого Vivaldi. Радует, что уже на этом этапе разработчики позаботились о сборках для Linux, а в обсуждении грядущих версий прислушиваются к мнению пользователей. **LXF**



»Окно настроек Vivaldi. Опций пока меньше, чем в старой Opera.



По рецептам доктора Брауна

Д-р Крис Браун

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

Эзотерическое системное администрирование из причудливых заворотов кишок серверной

Компьютер 3М

Свой первый графический рабочий стол (и BSD UNIX) я увидел на рабочей станции Sun-2. У меня в офисе была одна такая. В то время (примерно 1985 г.) ходили разговоры о компьютере '3М' — миллион байт памяти, миллион операций в секунду и миллион пикселей на экране. Иногда добавляли четвертую М — стоимость должна была не превышать один мегацент (\$10000). Моя Sun-2 еле-еле втискивалась в эти четыре М.

Перенесемся в настоящее. У недорогого ноутбука сегодня в 4000 раз больше памяти (4 Гб). Быстродействие процессоров сравнивать сложно, так как у современных процессоров несколько ядер и усложнились конвейеры команд, однако для скромного Intel Core i3 2000 млн команд в секунду — вполне реальная цифра. А цена (около \$250) упала в 40 раз. Но вот миллион пикселей — всё еще рабочее число; видимо, потому, что это уже почти порог зрительного восприятия.

Моя верная рулетка «Стэнли» говорит, что от меня до экрана ПК (340 мм шириной) — полметра. Можно посчитать, что экран занимает 38° (2280 минут) моего поля зрения. Человек со стопроцентным зрением различает два пикселя, находящиеся на расстоянии одной минуты дуги, поэтому по ширине экрана я должен различить около 2000. Если сильно увеличить их число, я просто ничего не увижу. Зрение у многих людей выше 100%, и, возможно, когда-нибудь появятся экраны '4К'. Но чтобы заставить меня купить что-то еще большее, понадобится очень одаренный маркетолог.



Не заходит Солнце

Солнце Sun несколько лет назад окончательно скрылось за горизонтом Oracle, но его лучи все еще освещают пейзаж Linux.

В этом месяце мне хочется окунуться в историю и подумать, откуда возникли некоторые технологии, применяемые в Linux. И на этой дорожке вы обязательно столкнетесь с Sun Microsystems.

Sun, пожалуй, лучше всего известна своим «железом», и в частности, профессиональными многоядерными серверами на основе архитектуры SPARC, предназначенными для крупных дата-центров. Однако Sun также внесла огромный вклад в мир открытого ПО. Билл Джой [Bill Joy] (единственный из основателей Sun, с которым я встречался вживую) написал редактор *Vi* и оболочку *C*, а также активно участвовал в разработке BSD Unix, особенно стека TCP/IP. Sun создала NFS (Network File System — Сетевая файловая система), которая все еще является основой для разделения файлов в сетях Unix и Linux. Она дала нам NIS и NIS+. В Sun работал Джеймс Гослинг [James Gosling], разработчик Java, и в конечном счете Sun выпустила ее под лицензией GPL. Компания также ввела в употребление PAM (Pluggable

Authentication Modules — Подключаемые модули аутентификации) в 1995 году; и они повсеместно используются для аутентификации в Linux. За счет приобретения других компаний Sun получила другие важные технологии, включая *MySQL*, *Open Office* и *VirtualBox*. Sun также дала нам ZFS — пожалуй, лучшую из файловых систем.

Несмотря на то, что OpenSolaris (2005) не был прямым предком семейства Linux, он стал важным рубежом в открытии кода Unix. Это был дар свыше для тех из нас, кому нужен был бесплатный «настоящий» Unix, способный работать на обычных компьютерах. Как сказал Ричард Столлмен [Richard Stallman], «Мне кажется, что из всех компаний именно Sun больше всего дала сообществу свободного ПО в виде ПО. Это лидерство. Это пример, которому, надеюсь, последуют другие». К сожалению, после того, как в 2010 г. Oracle купила Sun, было решено прекратить открытую разработку ядра и вместо этого выпустить проприетарный дистрибутив под названием Solaris Express. Прямо перед закрытием исходного кода Solaris сообщество разветвило проекты Illumos и OpenIndiana, чтобы мечта не умерла. (Условно говоря, Illumos — это проект ядра, а OpenIndiana создает дистрибутив на его основе.)

Будучи куплена Oracle, Sun перестала существовать как компания, оставив нас в тревоге о будущем многих открытых проектов, хранителем которых она была.



Яблочная Lisa

В компьютере Lisa, появившемся в 1983 г., использовались идеи Xerox Alto, и он был первым персональным компьютером с графическим рабочим столом, который поступил в коммерческую продажу. Наличие всего семи характерных приложений (*lisawrite*, *lisadraw*, *lisacalc*, *lisagraph*, *lisaproject*,

lisa!ist и *lisaterminal*) не давали усомниться в предполагаемой аудитории данного компьютера, который, однако не обернулся большим коммерческим успехом для компании Apple, став (в исторической перспективе) всего лишь очередным этапом на пути к победоносному Macintosh.

Сказ о компьютерной мыши

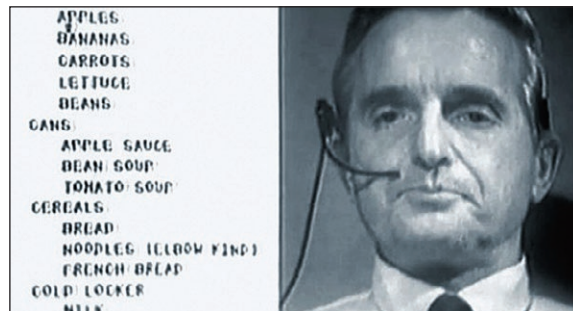
Отслеживаем историю компьютерной мыши с 1968 года и наслаждаемся выдающейся презентацией Дугласа Энгельбарта.

У меня для вас вопрос. Кто придумал окна, мыши, меню и т.д.? Чур, не гуглить... Что вы говорите? Microsoft? М-мм... нет. Microsoft — превосходный продавец, но... (я рискую открыть шлюзы для очередной войнушки) придумала она очень немногое. Ну, может, тогда Apple? Это немного ближе к истине. Apple определенно была первой компанией, которая предложила эти технологии большому количеству пользователей, сначала с компьютером Lisa 1983 года выпуска и позднее с MAC Classic в девяностых. Lisa продавался примерно за \$10000, что делало его доступным некоторым бизнес-пользователям, а цена на Classic опустилась до \$1000, сделав его по-настоящему «персональным» компьютером. Но концепция рабочего стола с окнами, иконками, меню и мышью появилась не в Apple.

Оказывается, Стива Джобса и его команду из Apple немало впечатлил визит в исследовательский центр Херох в Пало-Альто в 1979 г. — там они увидели компьютер под названием Херох Alto, который считается первым компьютером с графическим рабочим столом. (Кстати, в Херох PARC также изобрели Ethernet, значение которого тоже было огромным, но он не относится к нашей истории.) Однако история мыши уходит еще дальше, потому что на ребята из Херох PARC, в свою очередь, повлияли разработки команды, возглавляемой Дугласом Энгельбартом [Douglas Engelbart].

Энгельбарт работал в Стэнфордском исследовательском институте в группе с интересным названием — Центр исследования роста (ARC). Энгельбарт хотел вывести компьютер за границы чисто вычислительных задач и сделать его «инструментом, который поможет людям в работе со сложными информационными структурами». Я написал короткую историю о том, что может произойти, если мы позволим этому зайти слишком далеко [см. «Рецепты доктора Брауна», стр. 60 LXF191/192]. И, что важно для нашей темы, именно Энгельбарт считается изобретателем компьютерной мыши.

А теперь немного сменим ракурс. Для человека, на жизнь которого пришлось активные технологические прорывы, это может показаться странным, но иногда я чувствую разочарование



» Дуглас Энгельбарт и экран его компьютера в Матери всех презентаций (1968 г.).

от того, что не стал свидетелем некоторых ключевых исторических моментов: не включил первую электрическую лампочку; не наблюдал первой передачи данных через Атлантику; не настроил свой детекторный приемник, чтобы впервые услышать слабый голос, по чистому волшебству пробивающийся через эфир. И еще: не посетил Мать всех презентаций (как ее впоследствии называли), представленную Энгельбартом и его коллегами из ARC 9 декабря 1968 г. в Сан-Франциско.

Презентация полностью (за исключением кусочков, утерянных из-за смены бобин!) доступна на YouTube [www.youtube.com/watch?v=JflqzSoTMOs]. Качество изображения ужасное, но, сумев с ним смириться, вы увидите интерактивный редактор с различными иерархическими представлениями текста, выделением, удалением и вставкой текста с помощью мыши и простым вариантом гиперссылки. Вы даже увидите Энгельбарта, демонстрирующего иерархическое представление самой презентации на компьютере — и это за 22 года до PowerPoint.

Коллега Энгельбарта переворачивает мышью, чтобы показать, как она работает; тогда у нее было два колесика, расположенных под прямым углом друг к другу. (Первую мышью изобрел Билл Инглиш [Bill English], а до мыши с координатным шариком он дошел только в 1972 г. Оптические мыши появились гораздо позже, в 1980 г. Они были гораздо удобнее — их не приходилось постоянно разбирать, чтобы доставать оттуда волосяные комки и крошки печенья. В презентации Энгельбарт объясняет концепцию курсора (он называет его «следящей точкой»), который отмечает положение мыши на экране. Он также показывает пятиклавишное сенсорное устройство, которое позволяет вводить символы, зажимая пальцами «аккорды». В презентации есть и впечатляющая демонстрация сети — Энгельбарт в реальном времени взаимодействует с компьютером, который находится на расстоянии 30 миль от него в Менло-Парке.

Энгельбарт скончался в июле 2013 г., и, как и после смерти Денниса Ричи [Dennis Ritchie], пресса (по большей части) совершенно упустила возможность написать некролог гению. Сегодня, когда у нас есть сенсорные экраны с мультитач, достаточно легкого движения руки, и компьютер поймет ваше желание. Но не забудем, что всё это началось 50 лет назад, когда Дуглас Энгельбарт поместил в коробку два колесика под нужным углом друг к другу, перевернул ее и назвал это мышью.

»



» Один из прототипов мыши Энгельбарта. Фанаты Apple, как видите, здесь всего одна кнопка!

ClamAntiVirus

Сканирование файловых серверов и почтовых шлюзов на наличие вредоносного ПО помогает избежать попадания зараженных файлов в Windows.

Если у вас Windows, то обойтись без антивирусной защиты нельзя, и у большинства пользователей, как я подозреваю, есть подписка на приличный антивирус. Linux, хотя и не совсем лишен вредоносного ПО, страдает от него в гораздо меньшей степени. Аналитики обычно объясняют это двумя факторами. Во-первых, модель безопасности в Linux более прочная, что сильно затрудняет написание вирусов. Во-вторых, низкая рыночная доля Linux делает его менее доходной целью. К этим двум факторам я бы добавил третий: пользователи Linux более внимательны и искушены в вопросах безопасности. Например, они могут устанавливать только подписанные цифровой подписью пакеты из репозитория своего дистрибутива. Не знаю, насколько каждый из трех факторов имеет место в реальности, но ясно одно: Linux не является тем «низко висящим фруктом», который предпочитают авторы вредоносного ПО. Так есть ли смысл вообще запускать антивирусное ПО в Linux? Ну, сканировать каждый исполняемый файл при обращении к нему, пожалуй, вряд ли стоит, но если в вашей системе Linux есть файлы, которые затем попадут на компьютеры с Windows (например, если это почтовый сервер или файловый сервер *Samba*), то такие файлы определенно стоит сканировать на вирусы до переезда в Windows.

Clam AntiVirus — открытый (с лицензией GPL 2) антивирусный продукт, доступный для различных платформ, включая Linux, Solaris, FreeBSD и Windows. *ClamAV* принадлежал Sourcefire, которую в октябре 2013 года поглотила Cisco. Сейчас он активно поддерживается Talos Group, группой исследований в области безопасности и интеллекта в Cisco. Их сайт — <http://clamav.net>, а несколько архивов списков рассылки доступны на <http://lists.clamav.net>. Сердце антивируса — разделяемая библиотека; на ее основе работают сканер в командной строке и многопоточный демон сканирования. Одно из его основных назначений — сканирование входящих сообщений в почтовых шлюзах.

Я решил взглянуть на *ClamAV* в Ubuntu 14.04. Установить его легко, поскольку он есть в репозиториях Ubuntu:

```
$ sudo apt-get install clamav clamav-docs
```

Для разрешения зависимостей также были установлены пакеты *clamavbase*, *clamav-freshclam* и *libclamav6*.

Основной пакет *clamav* содержит три исполняемых файла: *clamscan*, *sigtool* и *clambc*, с соответствующими man-страницами.

Примемся за дело и попробуем выполнить сканирование с командной строки с помощью *clamscan*:

```
$ sudo clamscan -i -r Training/
Training/XXX/tmp.tar: Trojan.Linux.RST.b FOUND
Training/XXX/psybnc/whiper: Linux.RST.B-1 FOUND
Training/XXX/psybnc/pico: Linux.RST.B-1 FOUND
----- SCAN SUMMARY -----
Known viruses: 3720060
Engine version: 0.98.5
Scanned directories: 476
Scanned files: 3570
Infected files: 3
Data scanned: 1771.99 MB
Data read: 1168.35 MB (ratio 1.52:1)
Time: 277.079 sec (4 m 37 s)
```

Здесь *-r* означает рекурсию, а *-i* — вывод отчета только для зараженных файлов. Я сократил пути до 'XXX', чтобы они уместились в нашу журнальную колонку, но сам отчет настоящий. Три указанные здесь файла сначала меня тревожили, пока я не понял, что это примеры из курса по безопасности Linux, который я когда-то читал. Как видите, по умолчанию *clamscan* просто выводит на экран имена зараженных файлов; но можно попросить и отправить их в «карантин», то есть в заданный каталог, или просто удалить. Только подумайте, что мы сделали: одной короткой командой установили утилиту и инициализировали базу данных вирусов, другой короткой командой запустили сканирование, и никто не предложил ничего купить. Разве Linux не прекрасен?

Агент Clamd, лицензия на сканы

Также есть и графический клиент под названием *clamtk*. По сути, это всего лишь оболочка вокруг *clamscan*, не добавляющая никаких функций, но он чуть больше похож на антивирусный сканер в Windows. Сканирование в *ClamAV* также доступно в виде демона (*clamd*), который распространяется в отдельном пакете (*clamav-daemon*). По умолчанию этот демон слушает доменный сокет Unix, поэтому обратиться к нему можно только с локального компьютера. Однако его можно настроить так, чтобы он слушал порт TCP, что дает возможность запустить *clamd* как «агента» на всех своих компьютерах и управлять сканированием из одного центрального пункта. Существует клиентская утилита *clamdscan*, которая подключается к *clamd*; однако, насколько я понял, подключить его к удаленному демону нельзя. Для этого понадобится добыть программу на Python под названием *clamdscan.py*. Демон *clamd* записывает свои действия в файл, указанный в директиве LogFile в его файле настроек */etc/clamav/clamd.conf*.

Если вы хотите написать утилиту для сканирования с нуля, то есть библиотека (*libclamav*), которая содержит сам движок для сканирования. Эта библиотека хорошо задокументирована в руководстве пользователя *ClamAV*.

В руководстве пользователя упомянуто сканирование «при обращении» с помощью модуля *dazuko*: это сторонняя библиотека ядра, которая перехватывает вызовы доступа к файлам и передает информацию о файлах пользовательскому приложению. Идея состоит в том, чтобы поддерживать антивирусное сканирование «при обращении», журналирование доступа к файлам или другие внешние утилиты безопасности. Правда, официальный сайт <http://dazuko.org> выглядит заброшенным; последний



► В поисках вирусов *ClamAV* может проникать внутрь самых разных форматов файла.

релиз был выпущен почти четыре года назад, и в данный момент проект не поддерживается. Однако есть файловая система пространства пользователя на основе FUSE для Linux под названием *clamfs*, которая предоставляет антивирусное сканирование файлов при обращении через *clamd* (см. врезку «Clamfs» внизу).

База данных сигнатур вирусов

Качество антивирусного продукта зависит от двух факторов: от свежести базы сигнатур вирусов и от количества поддерживаемых форматов файлов. Сигнатуры вирусов в *ClamAV* хранятся преимущественно в двух файлах (**daily.cld** и **main.cvd**) в каталоге **/var/lib/clamav**. Для обновления этой базы данных вручную можно запустить программу *freshclam*:

```
$ sudo freshclam
ClamAV update process started at Wed Dec 31 06:30:56 2014
main.cvd is up to date (version: 55, sigs: 2424225, f-level: 60,
builder: neo)
Downloading daily-19859.cdiff [100%]
Downloading daily-19860.cdiff [100%]
daily.cld updated (version: 19860, sigs: 1299145, f-level: 63,
builder: neo)
bytecode.cvd is up to date (version: 244, sigs: 44, f-level: 63,
builder: dgoddard)
Database updated (3723414 signatures) from db.local.clamav.net
(IP: 193.1.193.64)
```

Вы увидите, что *freshclam* загружает только различия (diff), а не всю базу данных (как при первоначальной установке.)

Freshclam также можно запустить как демон (**freshclam -d**) для автоматического обновления базы данных. На самом деле, при установке *ClamAV* в Ubuntu демон *freshclam* автоматически настраивается на запуск раз в час, без необходимости дальнейших настроек. Лог-файл активности *freshclam* находится в **/var/log/clamav/freshclam**, и быстрый просмотр этого файла покажет, что обновления в базу данных сигнатур вирусов добавляются (и загружаются на мой компьютер) раза четыре в день. Для просмотра сигнатур вирусов используется команда **sigtool**. Например, для вывода всех сигнатур файла **daily.cld** скомандуйте:

```
$ sigtool --list-sigs daily.cld
Trojan.Hupigon-9863
Trojan.IRCBot-1971
Trojan.SdBot-8227
... плюс еще 3,7 миллиона ...
```

А для поиска конкретных сигнатур по регулярному выражению можно попробовать нечто вроде этого:

```
$ sigtool --find-sigs="Linux.*Worm" daily.cld
```

Как и полагается в открытом проекте, вы тоже можете добавить собственные сигнатуры вирусов, если найдете что-то, еще не распознаваемое *ClamAV*. Просто отправьте их по адресу community-sigs@lists.clamav.net.

Обычно сигнатуры вирусов не встречаются среди файлов системы в «чистом» виде. Скорее всего они будут прятаться в каком-то образе упакованных файлов (например, архивах или файлах MSI, собранных для установщика Microsoft) или файлах, сжатых, например, *Gzip*. Файлы Windows PE (Portable Executable — Портативные исполняемые файлы) — довольно частая мишень для вирусов — нередко упаковываются такими утилитами, как *UPX* или *Petite*, или намеренно шифруются программами вроде *Crypter* от Yoda, чтобы спрятать их от глаз антивирусных сканеров. Чтобы *ClamAV* мог найти вирусы везде, он обладает способностью



► Настоящий троянский конь, породивший цитату «Бойтесь гиков, GIFы приносящих». Так я и делал.

проникать в подлинно огромное количество форматов файлов — от архивов TAR, CPIO, GZIP и BZIP2 до PDF, файлов почтовых ящиков, архивов Windows Cabinet и массы сжатых или зашифрованных форматов PE. Пакет *clamav-testfiles* включает 44 файла различных типов, которые были «заражены» сигнатурой вируса ClamAV-Test-File и которыми можно воспользоваться для тестирования. Неудивительно, что *clamscan* нашла все эти файлы. (Я также скопировал все эти файлы в Windows и просканировал их премиум-версией Norton 360. Он ничего не нашел, поскольку вирусной сигнатуры ClamAV-Test-File нет в его базе данных.)

Библиотека также включает модуль «защиты от утраты данных», умеющий находить в текстовых файлах номера кредитных карт крупнейших поставщиков кредитных карт, а также номера социального страхования США, но я не проверял его в деле.

Фильтрация почты

Как уже упоминалось, одно из основных назначений *ClamAV* — сканирование сообщений, получаемых почтовым шлюзом. Это значительно упрощается благодаря технологии под названием *milter* (mail filter — почтовый фильтр). По сути, *milter* представляет собой набор связок, с помощью которых агенты передачи почты, такие как *Postfix* и *Sendmail*, взаимодействуют с внешним антивирусным сканером или другой программой фильтрации на различных этапах доставки письма. Пакет *clamav-milter* содержит необходимый фильтр; при установке пакета для него автоматически настраивается *Sendmail*, а более подробные инструкции по его настройке для Ubuntu можно найти по ссылке <http://bit.ly/UbuntuMailFiltering>. Вообще-то *clamav-milter* — лишь один из многих доступных фильтров; полный список можно найти на www.milter.org. **LXF**

Clamfs

Clamfs — еще одна утилита, построенная вокруг *ClamAV*. Она реализует файловую систему пространства пользователя, все обращения к файлам в которой перехватываются *clamfs*, которая в свою очередь подключается к *clamd*, чтобы просканировать этот файл. Несмотря на отсутствие подходящей документации, мне удалось заставить это работать. Вот что нужно сделать:

1 Установите пакет *clamfs*:

```
$ sudo apt-get install clamfs
```

2 Скопируйте архив с примером конфигурационного файла в подходящее место и распакуйте его:

```
$ cd /etc/clamav
$ sudo cp /usr/share/doc/clamfs/
clamfssample.xml.gz clamfs.xml.gz
$ sudo gunzip clamfs.xml.gz
```

3 Задайте настройки в конфигурационном файле. Я изменил только строку **<filesystem>**, задав в ней путь к файловой системе, которую я намеревался

смонтировать, и каталог, в который решил ее смонтировать.

4 Запустите демон *clamfs*, указав путь к конфигурационному файлу:

```
$ sudo clamfs /etc/clamav/clamfs.xml
```

Теперь у меня появилась точка монтирования **fuse.clamfs**; любые файлы в ней при обращении к ним будут сканироваться. Если будет обнаружена вирусная сигнатура, то операция блокируется, и пользователю root отправляется письмо.

LINUX FORMAT

Подписывайтесь и читайте Linux Format на iPad или iPhone!

Доступно
в AppStore



А если у вас Android, подпишитесь на электронную версию журнала Linux Format на сайте shop.linuxformat.ru!

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Профи профи-инструментов не чурается

Фаррел: Вы это видели?
Мак-Клейн: А то! Я это сделал!
Крепкий орешек 4.0

Графическому дизайнеру по имени Сергей (jemmybutton в Живом Журнале) дали задание нарисовать иллюстрации к школьному учебнику-задачнику по физике. Это весьма специфичное задание. Иллюстраций нужно много, они в целом несложные, и в них множество повторяющихся элементов. Текст учебника набирался в LaTeX-формате, поэтому Сергей занялся поиском подходящей технологии, чтобы минимизировать свои затраты при сохранении качества результата. И мне чрезвычайно приятно, что он, как и я для аналогичной задачи, выбрал MetaPost.

Да, да, примерно десять лет назад я оформлял задачник по школьной физике. В отличие от меня, Сергей — профессионал, и то, что он выбрал в каком-то смысле полюбившийся мне инструмент — приятно. Это означает, что мое решение было оправданным, хотя в отличие от меня Сергей умеет рисовать. Я же мог только описывать картинки в виде кода.

Побочным результатом работы Сергея по проекту стал пакет для MetaPost fiziko, который он выложил под свободной лицензией GPLv3. А я жалею, что десять лет назад этого пакета не было, и радуюсь, что сейчас он появился. Иллюстрации для школьников по физике теперь можно сделать намного красивее! **PS** Закрывается Google Code. Бесплатное не значит свободное. E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Дрессировать черепах 70

Рисовальщик-черепашка помогает **Джонни Бидвеллу** строить пирамиды и растить фрактальные деревья, и все это — в дивном мире *Minecraft*.



Рисовать командами 74

Графические интерфейсы — ниже достоинства **Нейла Ботвика**: даже с изображениями он предпочитает работать из командной строки.



Улавливать движение 76

Береженого и Бог бережет, решил **Кент Ельчук**, и установил домашнюю систему видеонаблюдения. Перенимайте его опыт.



Управлять блогами 80

Блоги сейчас не ведет только ленивый (хотя их ценность для человечества не всегда очевидна). **Стивен Ву** научит вас создавать блог с собственной темой.



Работать с 1С 84

Михаил Вознесенский разбирает окопы Windows у практически монополюной в России программы для бухгалтерского учета. Отчеты будут по форме и в любимой ОС!

Код в учебниках

Строки исходного кода помещаются в цветные плашки. Если строка кода не помещается в колонке, ее остаток переходит на следующую строку, в той же плашке:

```
procedure TfrmTextEditor.  
  mniWordWrapClick  
A плашки разделены зазорами:  
begin  
  mniWordWrap.Checked := false
```

АКАДЕМИЯ КОДИНГА



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

Python 3 88

Не очень-то охотно переходят программисты на Python 3, но **Джонни Бидвелл** считает, что настала пора это сделать. Нарботки не пропадут.

Julia 92

Михалису Цукалосу приглянулся быстрый и динамичный язык для технических вычислений, и он выстраивает фундамент для программирования на нем.

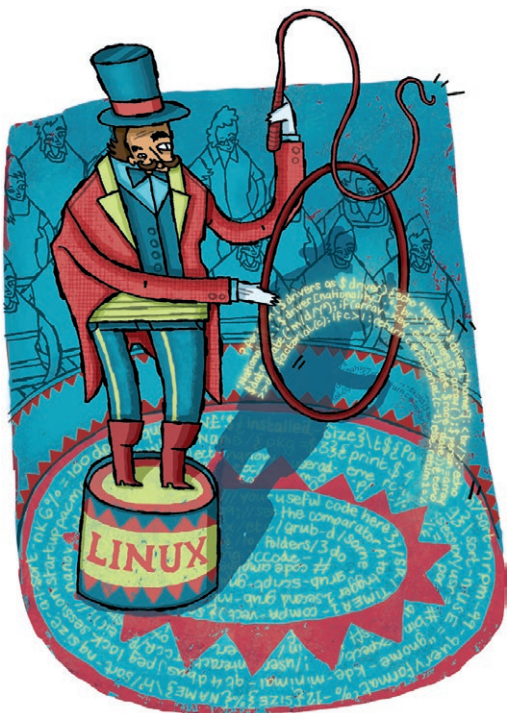
Minecraft/Pi: Черепахография

Джонни Бидвелл снова запускает *Minecraft* на Raspberry Pi и забавляется с графическими талантами, сокрытыми внутри скромной черепашки.



Наш эксперт

Джонни Бидвелл выяснил, что в Северной Америке слово «черепаха» описывает сухопутных, водяных и морских черепах, т.е. весь отряд черепаховых. В Великобритании значение более узкое.



Возможно, вы и стары, но еще не растеряли всю память, чтобы вспомнить программирование черепашкой графики на языке под названием Logo. На самом деле, языков с таким названием было несколько; первый из них появился в конце 1960-х. Как бы то ни было, с помощью захватывающей среды управления рисующей черепашкой (автоматизированной или абстрактной) пользователи могли с помощью простых команд рисовать фигуры и шаблоны из линий разных цветов. У черепашки была голова (или, на худой конец, хвост), и все команды (вперед, назад, на двадцать градусов влево и т.д.) применялись относительно текущего положения этой тварюшки, а не абсолютных экранных координат. Для лучшего понимания эффекта каждой команды учеников поощряли к «переволплощению»: им следовало вообразить себя покорной рептилией. Овладев основами, можно было изучить функции и передачу параметров, а уж затем лихо рисовать одной командой правильный объемный супер-пятиугольник (который, существуя он на самом деле, был бы 115-берным) с любой заданной длиной ребра.

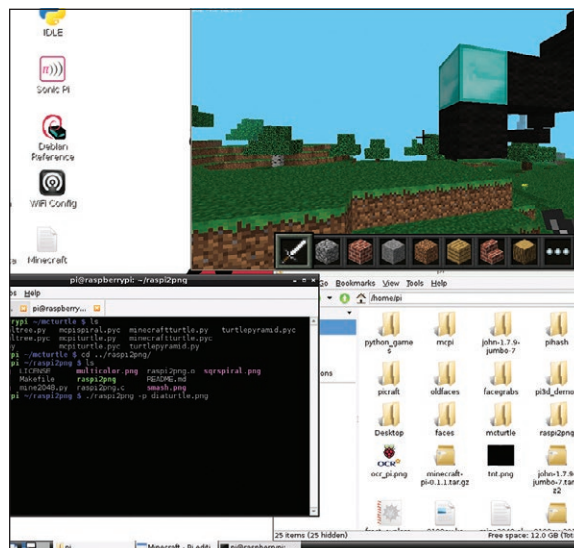
В Python есть собственный модуль черепашкой графики, с легковесным набором графических инструментов Tk. Если ваш дистрибутив — Raspbian, то этот модуль у вас уже установлен; и если вы никогда не вживались в роль рисующей черепашки, определенно стоит попробовать — загляните в документацию на <https://docs.python.org/2/library/turtle.html>. Однако на нашем

уроке мы внесем черепашку графику в *Minecraft*, то есть создадим свою реализацию Turtle. Ту, в которой наш панцирный принтер сможет ползать не в двух, а в трех измерениях. Я сказал «создадим свою»? Ну, не совсем чтобы свою: все это уже проделал наш старый друг Мартин О'Хэнлон [Martin O'Hanlon]. Зайдите на его сайт www.stuffaboutcode.com, и вы узнаете все о разнообразных проектах *Minecraft: Pi Edition*. Ранее мы уже «одолживали» код Мартина, и вы, возможно, видели его фантастически удобный класс `MinecraftDrawing` в предыдущих руководствах [см. Учебники LXF186, стр. 84]. В этом классе предусмотрены функции для рисования произвольных точек, линий, окружностей и сфер в мире *Minecraft*, и его можно принять за основу для нашего проекта `mcspirturtle`.

Прежде чем начать, стоит объяснить некоторые из трудностей, возникающих из-за перемещений нашей черепашки в трехмерном мире. В двумерном мире нас волновал только один угол: он определял, на сколько градусов влево или вправо от текущего направления нужно повернуть. При переключении в трехмерный мир у нас появляется еще одно направление, которое можно менять: угол подъема. Таким образом, у нашей черепашки будет четыре разных функции поворота, а также две функции перемещения (`forward()` и `backward()`). Мы будем отслеживать два направления с помощью свойств `heading` и `verticalheading` объекта `turtle`. Сначала такое кажется немного сложным, но стоит немного потренироваться — и это станет вашей второй натурой.

Скорая помощь

С помощью Libre-Logo можно воспользоваться черепашкой графикой в *LibreOffice*: зайдите в `View > Toolbars > Logo` [Просмотр > Панели инструментов > Logo]. Еще одна боковая дорожка для прокрастинации.



» По умолчанию наша черепашка представлена алмазным блоком, ибо она тверда.

Инспирируем спирали

В черепашкой графике легко рисовать фигуры, которые обычным образом создавать довольно трудно. Представьте себе спираль в двух измерениях: каждая координата задается параметрическим уравнением вида $x=a \cdot t \cdot \cos(t)$, $y=a \cdot t \cdot \sin(t)$, где t — параметр, который изменяется от 0 до бесконечности (или до тех пор, пока вам не опротивит рисование). Но черепашкам на тригонометрию наплевать, и если они хотят нарисовать спираль, то просто используют рецепт:

```
for step in range(100):
    t.forward(step // 2)
    t.right(45)
```

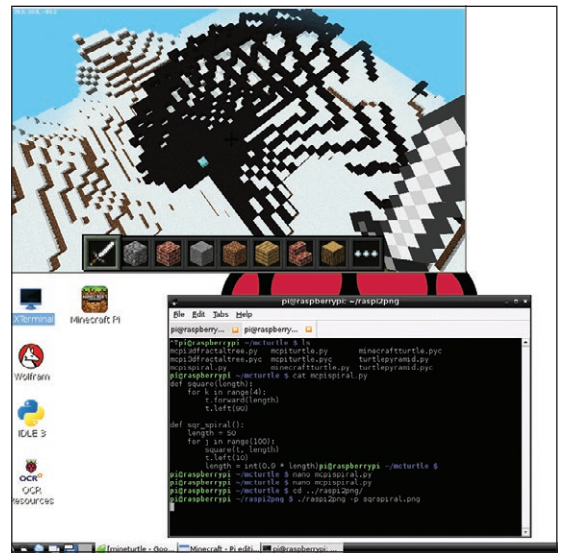
Да, спираль выходит не слишком гладкая, зато метод рисования гораздо проще. Вот еще один подход, при котором спираль создается извне путем поворота и масштабирования квадратиков:

```
def square(length):
    for k in range(4):
```

```
        t.forward(length)
        t.left(90)
def sqr_spiral():
    length = 50
    for j in range(100):
        square(t, length)
        t.left(10)
    length = int(0.9 * length)
```

Вызов второй функции, если все задано правильно, дает гораздо более впечатляющий результат, показанный на рисунке.

Если добавить еще одно измерение, все станет еще более импозантным: при «продавливании» окружности (выражаясь параметрически, окружность — это вырожденный случай спирали) мы получаем спираль; если же взять две окружности, расположенные одна напротив другой, получится двойная спираль. А раскрасив ее в четыре цвета, можно обогатить мир *Minecraft* моделью ДНК.



Наряду с шестью командами для относительного поворота и перемещения, наш объект `mcpi.turtle` принимает команды для установки абсолютного положения и направлений: `setx()` и т. д. для установки отдельных координат, `setposition()` для установки всех трех координат, `setheading()` и `setverticalheading()` для изменения углов направлений. Хотя эти команды не были частью исходного языка (в них нет смысла, если вы робот, не привязанный к экранным координатам), они очень удобны для перемещения в мире *Minecraft*. Также есть функции `penup()` и `pendown()` для управления тем, будет ли наша черепашка что-то рисовать, и `penblock()` для управления блоками, которыми она будет рисовать (по умолчанию используется черная шерсть [black wool]). Вызвав функцию `speed()` с числом от 0 до 10, можно изменить скорость черепашки: 10 — самая быстрая, 0 — самая медленная, а если указать 0, линии будут рисоваться мгновенно.

Пасем черепашку

Быть черепашкой в трехмерном мире, полном холмов, озер и других объектов, состоящих из блоков, по-своему нелегко. Представим на секунду, что вы черепашка и вам велено пройти на 50 блоков вперед; но вы уперлись в огромную гору. Как вы, существо послушное, поступите? Можно остаться на земле и обойти гору, или влезть на нее, или игнорировать красоту пейзажа и безжалостно прогрызть в горе ход. Похожая дилемма возникает, если вы стоите перед зияющей пропастью. Выбор между этими альтернативами осуществляется путем переключения черепашки (можете уже перестать представлять, что это вы) в режим `walk()` или `fly()`. Первая функция оставляет черепашку на уровне земли, поэтому координата y (и любые ее пути) будут изменяться. В режиме полета (по умолчанию) координата y останется постоянной, и черепашка будет летать над долинами и проделывать туннели в горах.

Установив *Minecraft* и `mcpi.turtle.py` [см. стр. 73 внизу], начнем исследовать возможности черепашки.

Загрузив *Minecraft*, откройте LXTerminal, перейдите (`cd`) в каталог, куда вы скопировали `mcpi.turtle.py`, и запустите Python (при желании можете использовать среду разработки IDLE).

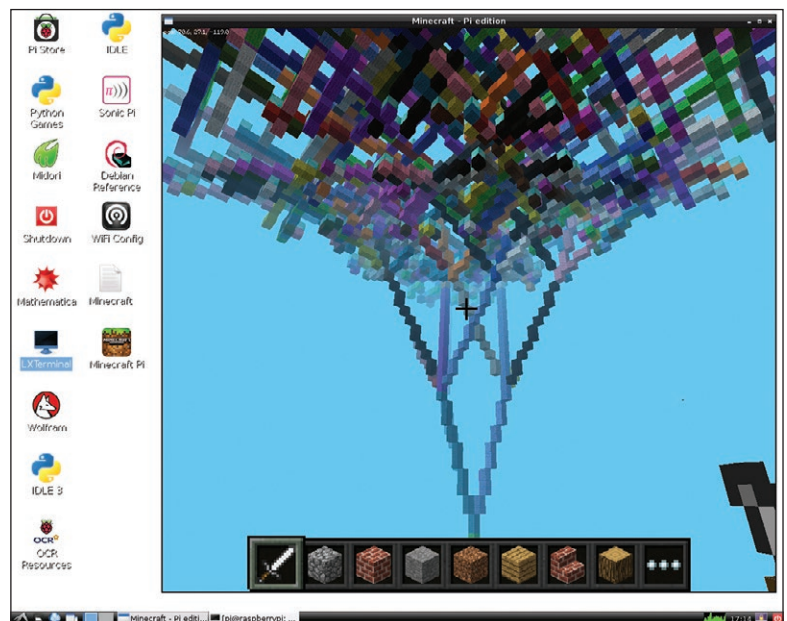
Импортируйте модуль и установите соединение с сервером игры, командами:

```
import mcpi.turtle.py
import mcpi.minecraft as minecraft
import mcpi.block as block
mc = minecraft.Minecraft.create()
```

И создайте новый объект черепашки в текущем положении Стива:

```
>>> pos = mc.player.getTilePos()
>>> t = mcpi.turtle.MinecraftTurtle(mc, pos)
```

Э, в вашем текущем местоположении появился алмазный блок! Да это ж наша черепашка. Желая переделать ее внешний вид, измените атрибут `turtleblock`. Например, чтобы превратить ее в каменный [stone] блок, сделайте следующее: »



» Эти психоделические деревья могут стать вашими всего за несколько строк кода.

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Скорая помощь



Если ваша черепашка перемещается в отрицательном направлении оси Y (которое вульгарно называют «вниз»), то блок-алмаз, который представляет черепашку, сотрет свою предыдущую отметку. Вы можете исправить эту ошибку, кашируя и восстанавливая блоки при движении черепашки.

```
>>> t.turtleblock = block.Block(block.STONE.id)
    Удовлетворившись видом черепашки, можно будет нарисовать что-нибудь высокохудожественное... ну, или не очень высоко. Следующий код нарисует грани пирамиды с квадратным основанием:
>>> for j in range(4):
>>> t.forward(20)
>>> t.right(90)
>>> t.right(135)
>>> t.up(52)
>>> t.forward(23)
>>> t.down(104)
>>> t.forward(23)
>>> t.up(52)
>>> t.right(135)
>>> t.forward(20)
>>> t.right(135)
>>> t.up(52)
>>> t.forward(23)
>>> t.down(104)
>>> t.forward(23)
```

Обратите внимание, что наша черепашка рисует под своим текущим положением. Это способствует интересной ошибке (см. «Скорая помощь» слева). А пока наслаждайтесь нашим примером, в котором мы создали каркасную модель пирамиды. Если вы заполните пробелы (или заставите тысячу-другую рабов сделать это за вас), то получится достойная дань фараону Стивхотепу из страны *Minecraft* (или нечто вроде иллюстрации внизу). Мило, но довольно нудно (а ведь нам даже не пришлось мучиться с геометрией), и не столь впечатляет, как пирамиды в Гизе (правда, пропорции близки: у пирамиды Хеопса в Гизе угол при основании чуть менее 52 градусов).

Вот еще более красочный пример: на сей раз мы засунем наши команды рисования в функцию. Эти команды можно либо напрямую ввести в интерпретатор Python, либо сохранить в файле, скажем, **ngon.py**, в том же каталоге, что и **mcpiturtle.py**.

```
import random
def ngon(t, n, length):
    angle = 360./n
    for j in range(n):
```

```
        colour = random.randint(0, 15)
        t.penblock(35, colour)
        t.forward(length)
        t.right(angle)
```

Теперь импортируйте модуль **ngon**, если сохранили его как внешний, переместите черепашку, чтобы убрать ее с дороги пирамиды, и в том же интерпретаторе запустите:

```
>>> import ngon
>>> ngon.ngon(t, 9, 10)
```

Они нарисуют правильный девятиугольник с ребром длиной в 10 блоков. С помощью модуля **random** мы изменяем цвет рисования черепашки. Тип блока у шерсти [wool] равен 35 (доступа к классу **block** у нас нет, поэтому мы не можем использовать более описательные названия, как раньше), и во второй строке кода устанавливается параметр **blockData**, который контролирует цвет блоков шерсти.

Lignum fractus

[игра слов: *лат. lignum fructus* — плодородное дерево, — прим. ред.] Мы уже рассказывали о фракталах, в последний раз создав снежинки фон Коха [von Koch] в *GIMP* с помощью Python [см. Учебники **LXF190**, стр. 88]. Затем мы узнали все о рекурсии, и как с помощью функции, вызывающей самое себя, получить красивые картинки. На нашем уроке мы сосредоточимся скорее на ботанике, чем на метеорологии, и нарисует несколько замечательных деревьев, после чего наш учебник не будет выделять в окружающую среду углекислый газ.

Нарисовать двумерные фрактальные деревья с помощью черепашкой графики довольно просто — начните с прямой линии (ствол), поверните направо под установленным углом и повторите процедуру для более короткой линии, затем поверните налево под установленным углом и повторите процедуру.

В Python общая форма этого кода (включая первоначальные установки) будет такой:

```
def tree2d(branchlen, t):
    if branchlen > 2:
        t.forward(branchlen)
        t.right(20)
        tree2d(branchlen — 2, t)
        t.left(40)
        tree2d(branchlen — 2, t)
    # тогда вернуться к исходному направлению/положению
    t.right(20)
    t.backward(branchlen)
```

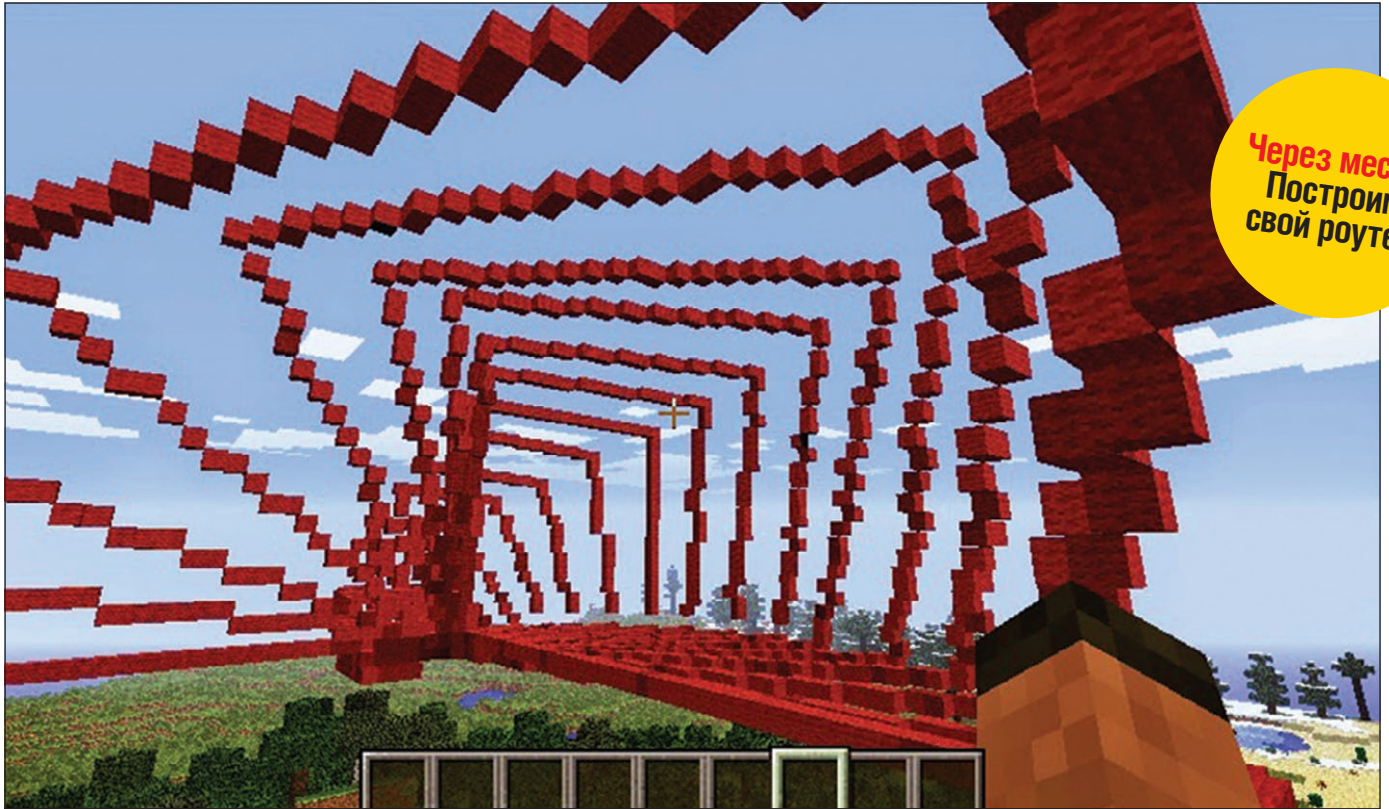
Затем вызов функции **branchlen(20)** будет медленно, но верно выращивать нам дерево. Выражение **if** позволяет избежать превращения функции в бесконечную рекурсию — с каждым разом мы рисуем ветки всё более короткие, но функция откажется что-либо делать, если вызвать ее с длиной ветки в 2 или меньше. Поэтому после нашего вызова рисуется дерево с длиной ствола 20 и ветками длиной 4.

Обобщить эту процедуру на случай трех измерений довольно просто — вместо двух вызовов рекурсивной функции у нас будет четыре: по одному для каждого направления компаса и по одному для поворота, отклоняющегося от родительского вертикального направления на 20 градусов. Дополнительно можно придать нашему дереву некий романтический флер, нарисовав несколько веток покороче (вверху) различных цветов — как символ весенних побегов, которые не сегодня-завтра расцветут. Ветки покрупнее (и ствол) будут одеревенелыми, как и в природе. Наконец, поскольку алгоритм способен изрядно тормозить (вызов **tree(20)** провоцирует 4096 рекурсивных вызовов), мы сбросим исходное



» «Стив, что ты видишь?» «Удивительные штуки!» Или, быть может, древние проклятия и болезни, заложенные в блоки. [Картинка с сайта raspberrypi-spy.co.uk]

» Подпишитесь на печатную и электронную версии на www.linuxformat.ru/subscribe!



Через месяц:
Построим
свой роутер

положение черепашки напрямую вызовом `t.setposition()` вместо `t.backward()` и всяких ненужных вычислений и перемещений внутри блока.

Ну и, без дальнейших предисловий, вот наша функция трехмерного дерева:

```
def tree(branchLen,t):
    if branchLen > 6:
    if branchLen > 10:
        t.penblock(block.WOOD)
    else:
        t.penblock(block.WOOL.id, random.randint(0,15))
        x,y,z = t.position.x, t.position.y, t.position.z
        t.forward(branchLen)
        t.up(20)
        tree(branchLen-2, t)
        t.right(90)
        tree(branchLen-2, t)
        t.left(180)
        tree(branchLen-2, t)
```

```
t.down(40)
t.right(90)
tree(branchLen-2, t)
t.up(20)
t.setposition(x, y, z)
```

Все это вместе со всеми необходимыми настройками вы найдете в файле `mcp3dfractaltree.py` на **LXF DVD**. После импорта этого файла (во время загрузки мира *Minecraft*) анимация отключится (по причине вызова `speed(0)`), и в вашей текущей позиции примется расти довольно занятное дерево.

Если получившееся дерево не придется вам по вкусу, его легко изменить. Ради эксперимента с трехмерными деревьями, почему бы не добавить к углам мелкие случайные отклонения? Ведь в природе не всё всегда идеально и симметрично. К тому же, деревья (мы говорим о Северном полушарии) имеют обыкновенные тянуться к югу, и это солнцепоклонничество можно учесть в вашем коде, ограничившись одним направлением. На чем очередная часть нашего замечательного сериала *Minecraft: Pi Edition* и завершается. **LXF**

► Спирали черепашкам нипочем. Через неделю: создадим в *Minecraft* собственную ДНК для ГМО! [Ред.: — Щаз.]

Скорая помощь

Масса других примеров с черепашкой в *Minecraft* есть в GitHub Мартина: <http://bit.ly/MineCraftTurtle>.

Установка Minecraft:Pi Edition и модуля turtle

Мы уже рассказывали об этом, но поскольку появились пакеты для Raspbian, всё стало немного попроще. Во-первых, если вы загружали Raspbian не ранее сентября 2014 года, то у вас уже установлен *Minecraft*, и делать ничего не придется. А если нет, то команды

```
$ sudo apt-get update
$ sudo apt-get install minecraft-pi
```

выполняют всю необходимую работу. До появления этих пакетов требовалось установить программу в свой домашний каталог

и скопировать файлы API Python из домашнего каталога в каталог проекта.

Описанный подход все равно сработает, но новый вариант проще. На нашем DVD вы найдете файл `mcpiturtle.py`. Он будет работать без всяких в нем поправок, если вы установили пакеты как указано ранее, и будет работать и в старом варианте, если в каталоге `mcp` у вас есть файлы API, а в каталоге уровнем выше есть файл `mcpiturtle.py`. В противном случае потребуется изменить первые две строки `import` согласно

расположению файлов API. Проверить это можно, запустив *Minecraft* (`$ minecraft`) и войдя в мир. В отдельном окне терминала перейдите в каталог, куда вы скопировали `mcpiturtle.py`, и запустите свой проект командой

```
$ python ./mcpiturtle.py
```

Она вызовет тесты по типу

```
if __name__ == "__main__":
```

которые включают: рисование пятиугольника, изменение цвета пера и произвольную загогулину-меандр.

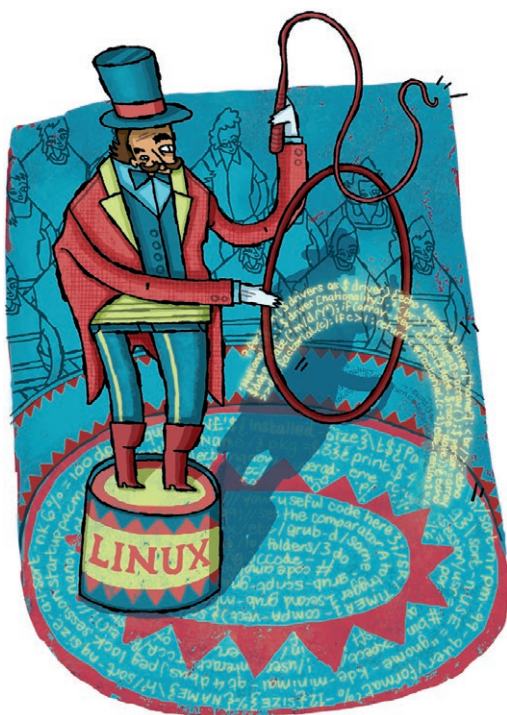
ImageMagick: Конвертируем

Нейл Ботвик призывает забыть о *GIMP*, прогнать мышь и научиться работать с изображениями из командной строки.



Наш эксперт

Нейл Ботвик обладает большим опытом загрузки компьютеров, ведь они у него по штучке в каждой комнате, а вот с перезагрузкой ему хуже, поскольку с Windows он перешел на Linux.

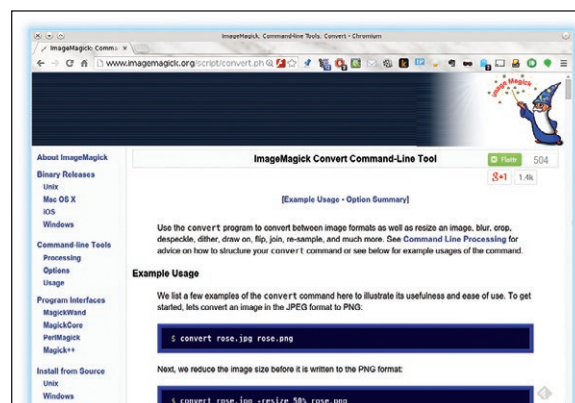


Попросите группу пользователей Linux назвать графический пакет, и подавляющее большинство первым делом упомянет *GIMP* (GNU Image Manipulation Program — Программа для управления изображениями GNU). Еще чуток голосов получают другие программы, но одному из самых используемых графических пакетов в Linux, *ImageMagick*, вряд ли перепадет много. Это набор программ для обработки изображений с командной строки; и в чем же их смысл, возможно, спрашиваете вы? Ведь редактировать изображения лучше всего в графическом редакторе? Если вы хотите подретушировать или еще как-нибудь изменить одно изображение — возможно, да; ну, а если нужно конвертировать в другой формат целый каталог с изображениями, или изменить их размер? Здесь и приходят на помощь программы из пакета *ImageMagick*, и поэтому он очень часто используется в Linux, пусть даже и неявно.

Самая популярная команда пакета *ImageMagick* — `convert`, и преобразование форматов — простейшая из ее операций. Команда

```
convert pic1.jpg pic1.png
```

делает именно то, что вы и подумали: преобразует изображение из JPEG в PNG. Она создает новый файл; изменение файла с замещением мы рассмотрим чуть позже. Указывать типы входного и выходного файлов не нужно — `convert` сама распознает



» Если ман-страницы покажутся вам слишком краткими, на сайте *ImageMagick* есть более подробное описание с примерами команд.

входной формат (отчасти в этом и состоит магия), а выходной формат определит по расширению файла. Чтобы это изменить, укажите перед именем выходного файла формат с последующим двоеточием:

```
convert pic1.jpg bmp:pic1.gif
```

Эта команда создаст изображение в формате BMP с расширением GIF. Данная опция может показаться довольно бесполезной, если не учесть, что, как и многие другие программы, `convert` распознает имя файла “-” как стандартный ввод или вывод. Так, если вы хотите отправить данные изображения другой программе в формате PPM, сделайте следующее:

```
convert pic1.jpg ppm:- | другая_программа
```

Для некоторых форматов можно указать дополнительные параметры — например, качество JPEG:

```
convert pic2.png -quality 75 pic2.jpg
```

На ман-странице и в онлайн-документации `convert` таких параметров множество.

Изменение размеров изображений

После изменения формата, одна из самых популярных задач — изменение размеров изображений, и `convert` тоже такое умеет.

```
convert pic3.png -resize 50% pic3_small.png
```

Это самая простая операция изменения размера, когда к изображению применяется один коэффициент масштабирования. Чтобы задать разные коэффициенты масштабирования по осям X и Y, воспользуйтесь параметром `-resize 40x60%`. Хотя здесь всего один знак процента, он применяется к обоим коэффициентам. Если указано одно значение, оно считается шириной нового изображения в пикселях; высота вычисляется так, чтобы сохранились пропорции.

```
convert pic3.png -resize 1000 pic1000.png
```

```
convert pic3.png -resize x1000 pic1000.png
```

Во втором примере указано значение высоты — в формате «ширина×высота» с пропущенной шириной. Если указаны и ширина, и высота, будет выбрано максимальное изображение из двух возможных с сохранением соотношения сторон. Если вы хотите задать абсолютные размеры и игнорировать соотношение сторон, добавьте к размеру!

```
convert pic4.png -resize 1000x600 pic5.png
```

```
convert pic4.png -resize 1000x600! pic5.png
```

Первая команда сохраняет соотношение сторон, вторая — создает изображение размером ровно 1000×600 пикселей. Если вы создаете миниатюры более крупных изображений, то можете сделать так, чтобы более мелкие изображения, которые уже есть в списке, не увеличивались — при этом обычно получаются изображения хотя и большего размера, но более расплывчатые. Для этого добавьте > таким образом:

```
convert pic4.png -resize 1000x600\> pic5.png
```

> означает, что размер нужно менять только тогда, когда размер исходного изображения больше указанного. Учтите: при запуске команды в оболочке нужно экранировать этот символ или поместить его в кавычки — иначе он будет интерпретирован как оператор перенаправления. Операции можно объединить в один вызов, в этом случае они применяются по порядку; и можно, например, изменить размер изображения и тут же преобразовать его в другой формат:

```
convert pic6.png -resize 30% -quality 75 pic6a.jpg
```

Добавляем текст

Одна из интересных возможностей — добавление на изображение текстовой надписи, например:

```
convert pic7.png -gravity SouthEast -family Times -pointsize 30 -annotate +10+10 "Some text" caption7.png
```

Ключевым параметром здесь является **-annotate**, за которым следуют координаты *x* и *y* текста и сам текст. Также важен параметр **-gravity**: он описывает точку, от которой отсчитываются координаты. В этом примере текст рисуется на расстоянии 10 пикселей от правого нижнего угла изображения. Параметры **-family** и **-pointsize** определяют шрифт и кегль текста. Обычно рекомендуется использовать **-family**, а не **-font**: если указанный шрифт недоступен, то **-family** выберет наиболее близкий к нему, а не выдаст ошибку. Параметры *ImageMagick* применяются начиная с того места, в котором они появляются в командной строке, поэтому параметры **gravity** и **font** нужно указать перед **-annotate**, иначе они не будут применены (ну то есть если нет последующих операций с текстом: в этом случае параметры будут применены к ним). Чтобы добавить к изображениям надписи с датой их создания, можно воспользоваться командой **convert** совместно с командой **date** следующим образом:

```
convert pic7.png -gravity SouthEast -annotate +10+10 "$(date -r QR.png +%d-%b-%y\ %H:%M)" new7.png
```

Convert создает новый файл, но иногда нужно заменить существующий. Для этого в *ImageMagick* есть команда **mogrify**. Она работает так же, как **convert**, за исключением того, что перезаписывает входной файл измененным. Сравните две следующие команды:

```
convert pic8.png -resize 50% pic9_small.png
```

```
mogrify -resize 50% pic9.png
```

Так как оригинал будет уничтожен, то нужно быть уверенным, что выполняемая операция верна; поэтому ее стоит предварительно проверить командой **convert**. Однако у **mogrify** есть еще один туз в рукаве. С **convert** нужно указывать как входной, так и выходной файл, поэтому каждый вызов можно запустить только для одного файла. Зато **mogrify** требует только имя файла, и если скормить ей несколько имен файлов, команда выполнится для каждого файла по очереди. Так, команда

```
mogrify -resize 50% *.png
```

обработает все подходящие файлы в текущем каталоге. Мы сказали, что **mogrify** перезаписывает исходный файл, но здесь есть важное исключение: если команда использовалась для изменения формата файла, будет создан новый файл с соответствующим расширением, а старый останется нетронутым, даже если над ним выполняются и другие операции. Скажем, команда

```
mogrify -resize 25% -quality 70 -format jpg *.png
```

создаст JPEG-миниатюры всех PNG-файлов, найденных в текущем каталоге.

На нашем уроке мы лишь поверхностно коснулись возможностей пакета *ImageMagick*; у рассмотренных команд есть масса других параметров, и есть масса других команд. Например, **identify** делает то, что говорит, сообщая формат и размер файла, а с помощью команды **import** можно сделать снимки X-сервера, отдельных окон или всего экрана. Map-страницы довольно лаконичны, но на официальном сайте *ImageMagick* — www.imagemagick.org — есть масса объяснений и примеров. **LXF**

➤ Для получения подробной информации об изображениях любого типа воспользуйтесь командой **identify**.

Групповая обработка

Mogrify может работать с несколькими файлами, а **convert** обрабатывает только по одному файлу. Как же обработать каталог с файлами? Проще всего — с помощью оператора **for** скриптового языка оболочки.

```
for i in *.png
do
  convert "$i" -resize 500x500 "thumbnails/$i%.png|.jpg"
done
```

Этот цикл выполняет команды между **do** и **done** по одному разу для каждого файла,

соответствующего шаблону. Каждый раз **\$i** заменяется именем файла.

\$(i%.png) заменяется именем файла без расширения **.png** (% удаляет из имени следующую за ним строку), а затем мы добавляем расширение **.jpg**, поэтому фактически выполняются следующие команды:

```
convert "pic1.png" -resize 500x500 "thumbnails/pic1.jpg"
convert "pic2.png" -resize 500x500 "thumbnails/pic2.jpg"
...
```

Мы заключаем имена файлов в кавычки на случай, если в них попадутся пробелы, которые в противном случае могут сбить оболочку с толку. Это хорошо работает, если все файлы находятся в одном каталоге; если они находятся в подкаталогах, иногда проще воспользоваться **find**:

```
find photos -name '*.jpg' -exec convert "{}" -resize 25% "thumbnails/{}" \;
```

Каждый найденный файл передается команде, следующей за **-exec**, в которой **{}** заменяется именем файла. Опять же, мы вставляем кавычки, чтобы избежать проблем с пробелами.

Motion: Засечь и записать

Кент Ельчук показывает, как собрать систему видеонаблюдения из web-камеры и Raspberry Pi и как сохранить видео, записанное при обнаружении движения.



Наш эксперт

Кент Ельчук — опытный web-разработчик, энтузиаст Linux и создатель Cloner и CMS Sitemakin. Он любит экспериментировать с новыми способами улучшения Сети для каждого.



Предположим, что у вас нет ни одного пакета, необходимого для этого урока по видеонаблюдению и записи видео. Воспользуйтесь *Motion*, который является сердцем нашего урока. Наряду с ним вам понадобятся *Apache* (или *Nginx*) и PHP. Хотя наш урок ориентирован на Raspberry Pi, при желании сойдет и другой компьютер. Учтите, что если выберете *Apache* и PHP, всё пройдет относительно гладко, без добавочных изменений в настройках сервера и PHP.

Если вы решите выбрать *Nginx* вместо *Apache*, потребуется несколько лишних действий, а именно: установка *PHP-FPM*; смена

корневого каталога для файлов web-страниц; и редактирование файлов `/etc/nginx/sites-available/default`, `/etc/nginx/sites-enabled/default` и `/etc/php5/fpm/php.ini`.

Вкратце опишем каждый пакет. *Motion* используется для записи видео при обнаружении движения. Видеоролики будут записываться в виде Flash-файлов SWF. Впрочем, *Motion* позволяет просматривать изображение и без движения, как с видеочкамер в обычных системах видеонаблюдения.

Когда у вас появятся такие файлы, их нужно будет эффективно сортировать. Здесь в дело вступают web-сервер и PHP. С сервером *Apache* или *Nginx* к этим файлам можно обращаться через Интернет.

В реальности файлов будет очень много, и вы можете написать цикл на PHP для вывода ссылок на все файлы, по нажатию на которые ролики будут открываться во всплывающих окнах. В таком случае пригодится бесплатное приложение для всплывающих видео, такое как *Shadowbox*. На ваше счастье, код на **LXF DVD** содержит файлы, которые все это делают.

Таким образом подготовившись, вы получите систему для доступа к своим видеороликам. На нашем уроке мы покажем различные опции и расскажем, как воплотить их в жизнь. Поскольку такая система видеонаблюдения может использоваться у вас дома, имеет смысл защитить паролем все web-страницы и каталоги, где хранятся видеоролики. А на случай, если кто-то вломится в ваш дом и украдет или сломает Raspberry Pi, мы расскажем, как настроить резервное копирование видеофайлов на сторонний web-сервер, о существовании которого грабитель не будет даже подозревать.

Настраиваем систему

Поскольку наш урок посвящается *Motion*, то и установим первым делом его:

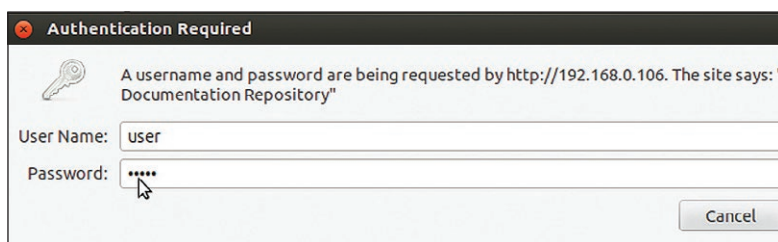
```
sudo apt-get update
sudo apt-get install motion
```

Разобравшись с одной установкой, добавим остальное: *Apache* —

```
sudo apt-get install apache2
sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
```

Продолжим и выполним ряд первичных процедур и проверок, чтобы убедиться: всё работает как полагается. Основные файлы, которые нужно изменить — `/etc/motion/motion.conf` и `/etc/default/motion`. Откройте `motion.conf` в любимом текстовом редакторе. Обычно предложенные далее параметры являются обратными по отношению к указанным по умолчанию. Например, “daemon off” превращается в “daemon on”:

```
daemon on
```



» Для ограничения доступа к файлам используйте простую аутентификацию по паролю.

```
webcam_localhost off
control_localhost off
```

Сохраните изменения, откройте файл `/etc/default/motion` и выполните следующие изменения:

```
start_motion_daemon=yes
```

Теперь пора провести тонкую настройку параметров. Нужно изменить частоту кадров, качество и минимальное количество кадров, необходимое для включения записи:

```
framerate 30
```

```
quality 90
```

```
minimum_motion_frames 5
```

Если не менять эти настройки, то два кадра в секунду — слишком мало: видеоролик будет прерывистым, и мы пропустим все действие. Поэтому изменим частоту кадров с 2 на 30. Второе изменение — очевидно, повышение качества. Третий параметр задает минимальное количество кадров движения, которое надо определить. По умолчанию оно равно 1. Если это число слишком мало, в результате у вас окажется масса ненужных записей, например, из-за мигания света. Помните, что у вас есть много других параметров и возможностей. С ними удобно познакомиться на официальном сайте (<http://bit.ly/MotionConfigFileOptions>).

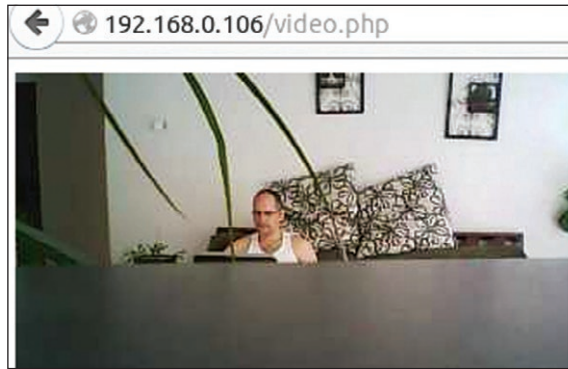
Может пригодиться, скажем, такая возможность, как снятие кадра с определенным интервалом, например, каждую секунду, каждую минуту или каждый час. Легко будет анализировать погоду или, например, понять, «не сидел ли кто-то у вас на диване и сломал его».

Настройка Motion

Подогнать все параметры под свои потребности очень просто, и в файле `motion.conf` немало простых и понятных комментариев, а на сайте и man-страницах есть более подробная информация. Естественно, без рабочей и совместимой web-камеры ничего не выйдет, и список web-камер, совместимых с Raspberry Pi, можно найти на http://elinux.org/RPI_USB_Webcams.

Использование камеры Plug-and-Play значительно упрощит эту задачу; дешевая доступная камера, которая нам подойдет — Logitech C170. Учтите: если вы пользуетесь Raspberry Pi, то камера Raspberry Pi не будет работать с Motion. Чтобы определить, подключилась ли камера, запустите команду `lsusb`.

Итак, у вас есть рабочая web-камера, рабочий web-сервер и подходящая конфигурация Motion. Это хорошо, но надо еще создать каталог для изображений, сделав Motion его владельцем.



► Полундра! Человек на диване! Запись видеопотока с камеры.

По умолчанию Motion сбрасывает изображения и файлы SWF в каталог `/tmp/motion`. Автоматически этот каталог не создается — это придется сделать вам:

```
cd /tmp
mkdir motion
chown motion:motion motion
```

Посмотрим, как все работает. Для начала запустите Motion командой

```
service motion start
```

а перезапустить его всегда можно командой

```
service motion restart
```

Первым делом проверьте и убедитесь, что все работает отлично: посмотрите, сможете ли вы открыть web-страницу по умолчанию. Поскольку у вашего Pi есть уникальный сетевой адрес, можете просто набрать его в браузере. Например, если Pi подключен к роутеру с IP-адресом 192.168.0.1, у вашего Pi будет IP-адрес вроде 192.168.0.106. Ну и URL тоже будет таким: <http://192.168.0.106>.

Если страница по умолчанию откроется, вы увидите сообщение о том, что все работает правильно. Если нет, вы увидите одну из типичных ошибок браузера, и станет ясно, что в порядке не всё.

Раз сервер работает, пора перейти к просмотру и записи видео. Видео можно проверить в браузере, набрав IP-адрес сети и порт. По умолчанию, для web-камер Motion используется порт 8081. И если набрать <http://192.168.0.106:8081> в браузере, вы должны увидеть свой видеопоток.

Несложная система вроде этой бывает очень удобной и не только для безопасности: например, можно приглядывать за младенцем, работая в другой комнате. Так как на данном этапе

«Подогнать все параметры под свои потребности очень просто.»

Скорая помощь

Если вы подключились по SSH и нужно отредактировать файлы, то воспользуйтесь Vim, в нем очень удобный поиск. Всё, что нужно для поиска — набрать `/` и затем то, что ищете. Для перехода к следующему результату просто наберите `n`.

Несколько web-камер

Одной web-камеры мало? Не проблема. В Motion добавить их легко. Откройте файл `/etc/motion/motion.conf` и задайте видеопотоки. Перейдя в конец файла, вы увидите различные строки комментариев, начинающиеся со слова "thread". Как видите, по умолчанию эти новые файлы располагаются в каталоге `/usr/local/etc`.

Для простоты можно изменить каталог с видеопотоками на `/etc/motion`. Тогда всё редактирование будет выполняться только в одном месте. Например, первый поток будет примерно таким:

```
thread /etc/motion/thread1.conf
```

Настроив потоки для всех своих камер, можно создать файлы настройки потоков. Файл для первого потока будет называться `thread1.conf`, для второго — `thread2.conf`, и т.д. Во все эти файлы настройки вам потребуется добавить всего по несколько строк.

Ниже приведены примеры для двух потоков. Как видите, у каждого потока есть собственный параметр `videodevice` [видеоустройство], произвольный текст, отображаемый с левой стороны кадра,

каталог для изображений и номер порта. Вот файл `thread1.conf`:

```
videodevice /dev/video0
text_left Camera #1
target_dir /var/www/images
webcam_port 8081
a вот — thread2.conf:
videodevice /dev/video1
text_left camera #2
target_dir /var/www/images_cam2
webcam_port 8082
```

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

с запущенным сервисом *Motion* все должно быть хорошо. Можете подойти к своей камере и немного попрыгать вокруг. (Вообще-то достаточно помахать рукой, но физические упражнения вам не повредят.) Затем, открыв новый каталог **motion**, вы увидите несколько файлов JPEG и как минимум один файл SWF.

Ну вот, у вас есть устройство для обнаружения движения, видимое в вашей локальной сети; можно пойти дальше и кое-что переделать, чтобы камеру можно было просматривать извне локальной сети. Для некоторых из вас именно такие настройки и нужны, особенно если ваша web-камера и Raspberry Pi хорошо спрятаны и в их работу вряд ли кто-то вмешается.

Однако мы дополним рассказ о том, как хранить данные в своем доме, объяснением, как скопировать файлы на другой сервер для безопасного хранения — на случай, если ваша SD-карта или жесткий диск дадут сбой или кто-то решит украсть, повредить или уничтожить вашу камеру (или камеры) в доме.

Подружим камеру с Интернетом

Идея в целом состоит в том, чтобы записывать, хранить и управлять вашим видео с сайта или указывая ваш IP-адрес, выданный провайдером. Чтобы узнать свой IP-адрес, зайдите на <http://whatismyipaddress.com>. Чтобы стала возможной трансляция видео, нужно настроить перенаправление портов на роутере, чтобы для IP-адреса вашей сети был доступен порт 8081.

Задно можно сделать то же самое для порта 80, поскольку тот же IP-адрес будет использоваться для отображения ваших web-страниц на компьютерах, находящихся за пределами локальной сети: например, на компьютере вашего друга на другом конце города или второй половинки где-то за океаном.

Выполнив указанные ранее изменения в настройках роутера, попробуйте набрать http://мой_ip-адрес_от_провайдера и http://мой_ip-адрес_от_провайдера:8081. Вы должны получить те же результаты, что и при проверке с локальным IP.

Следующий шаг — прибраться и просмотреть упорядоченные данные через URL вроде http://мой_ip-адрес_от_провайдера/video.php или с другого сайта с помощью `iframe`.

Чтобы добавить на страницу камеру со страницы `video.php`, нужно просто воспользоваться тэгом `` с IP-адресом сети и номером порта. Взгляните на код ниже, в нем это показано максимально подробно, а высота и ширина картинки соответствуют указанному в файле `motion.conf`:

```

```

Теперь представим сценарий, когда вы хотите транслировать видеопоток на другом сайте с помощью `iframe`. Всё, что вам нужно

сделать — вставить `iframe` в другую страницу, расположенную на другом сервере. Простая строка, которую нужно вставить, показана ниже.

```
<iframe style="width:320px; height:240px;" src="http://isp_ipaddress/video.php"></iframe>
```

Следующий фрагмент кода пояснит, как просмотреть файлы, записанные после обнаружения движения.

Сделав все это, двинемся дальше и найдем простой способ организации этих файлов. Первым делом надо настроить сохранение изображений и файлов SWF в web-каталоге. Корневой web-каталог находится в `/var/www/html` или `/var/www`. Сейчас все должно быть ясно, поскольку вы уже выполняли изменения в настройках *Motion*. Откройте `/etc/motion/motion.conf` и измените целевой каталог. По умолчанию это `/tmp/motion`. Новым целевым каталогом должен стать `/var/www/images`:

```
target_dir /var/www/images
```

Просмотр своего видео и изображений

Изменив файл `motion.conf`, наберите команду

```
sudo service motion reload
```

— и *Motion* будет записывать все новые файлы в каталог `/var/www/images`. Теперь вы сможете легко получить доступ к файлам, созданным сервисом *Motion*, и отображать их в Сети как любые другие web-страницы. Хотя мы изменили путь в `motion.conf`, каталог `images` еще не был создан. Сделаем это.

Каталог будет находиться внутри `www` или `html`. Если вам кажется, что мы повторяемся, значит, вы внимательны и знаете, что корневой web-каталог *Apache* может находиться по одному из двух путей:

```
cd /var/www
```

```
mkdir images
```

По умолчанию, владельцами каталога `www` будут пользователь `root` и группа `root`. Вы можете кое-что изменить: например, владельцами всех файлов назначить пользователя `pi` и группу `www-data`. Для этого сделайте следующее:

```
cd /var
```

```
chown -R pi:www-data www
```

Теперь нужно сделать так, чтобы этот каталог `images` был доступен для записи сервису *Motion*. У других файлов уже настроены подходящие владельцы и права доступа, но у каталога с изображениями — нет. Изменим их сейчас. Приведенный ниже фрагмент состоит из трех команд.

Первая команда добавит пользователя `motion` в группу `www-data`. Если вам интересно, `www-data` — существующие пользователь и группа для сервера *Apache*. Вторая команда дает права на каталог `images` пользователю `motion` и группе `www-data`, а последняя команда делает каталог доступным на запись, чтобы все изображения и файлы SWF волшебным образом появились в каталоге `images`:

```
usermod -a -G www-data motion
```

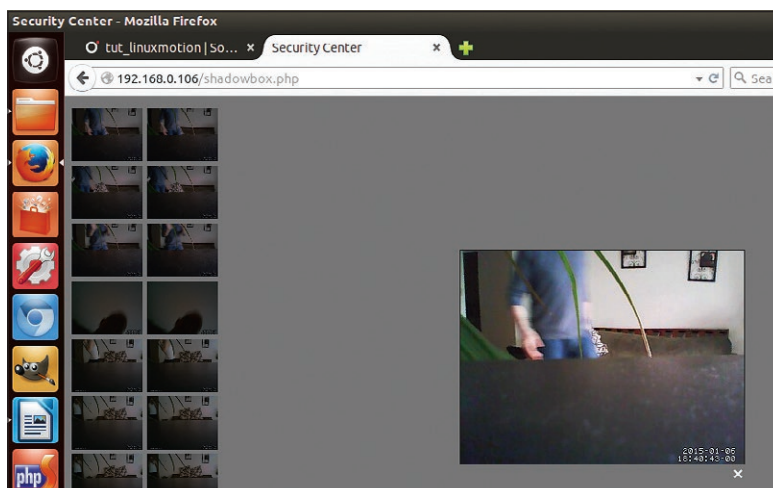
```
chown motion:www-data images
```

```
chmod 777 images
```

В каталоге `www` можно создать файл `shadowbox.php`, который будет использован для отображения картинки с камеры в сети. В этом файле — весь необходимый код для отображения миниатюры для каждого записанного файла SWF, самого видео и первое JPEG-изображение с начала движения.

Вывод содержимого в коде осуществляется так. Каталог с изображениями сканируется, создается массив файлов и сортируется по убыванию. Это несколько JPEG-файлов и один SWF-файл для каждого события. Имена всех файлов начинаются с события, за которым следуют дата и порядковый номер.

➤ **Отображение сохраненных изображений и Flash-видео.**



➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

Формат даты следующий: год, месяц, день, часы, минуты и секунды. После даты *Motion* добавляет порядковый номер, начиная с 01 (только для файлов JPEG).

Чтобы все оставалось простым и ясным, код в файле **shadowbox.php** будет отображать только одну картинку и один файл SWF для каждого события. В этом простом скрипте с помощью *shadowbox* также создаются всплывающие окна для первого файла JPEG и файла SWF для каждого события. Теперь вы видите запись всех движений до самого первого, записанного *Motion*. Этот файл выдает список всех результатов, и вам, скорее всего, захочется урезать его вывод.

Желая защитить web-страницы паролем через *Apache*, можете открыть файл `/etc/apache2/sites-available/default` и внести в него небольшие изменения. Найдите строку `<Directory /var/www/>` и добавьте три следующие строки под ней. Пример кода приводится ниже:

```
AuthType Basic
AuthName "Private Documentation Repository"
AuthUserFile /var/www/.htpasswd
Require valid-user
```

После этого перейдите в каталог `/var/www` и создайте пустой файл `.htpasswd`. Затем создайте имя пользователя и пароль простой командой, приведенной ниже. Пароль будет запрошен дважды. После ввода пароля нажимайте Enter:

```
sudo htpasswd /var/www/.htpasswd имя_пользователя
```

Так как довольно быстро накапливается много файлов и пространство на диске начнет таять, можно создать систему удаления старых файлов или копировать файлы на другой диск. Далее обсуждаются некоторые варианты копирования.

Варианты копирования

Для определения плана копирования следует посмотреть, сколько пространства будет обычно тратиться, и разработать план в соответствии с этими условиями. Просто! Например, если вам требуется 1 ГБ в неделю и у вас карта на 8 ГБ, то можно копировать на удаленный сервер TAR-архивы изображений через SCP и удалять все файлы старше недели. Поскольку имена файлов содержат год, месяц и день, определить старые файлы будет довольно легко.

В файле **purge.php**, который имеется на **LXF DVD**, реализован механизм такого удаления: в нем удаляется каждый файл, который старше определенного количества дней. Чуть позже я объясню код более подробно.

Сначала сканируется каталог **images**, и все файлы добавляются в массив. Затем этот массив файлов перебирается в цикле `foreach`. Для преобразования имен файлов в даты, которые могут быть использованы для сравнения, используются встроенные

```
73002 -rw-r--r-- 1 motion motion 17211 Jan 6 18:40
73001 -rw-r--r-- 1 motion motion 16824 Jan 6 18:40
73000 -rw-r--r-- 1 motion motion 16939 Jan 6 18:40
72999 -rw-r--r-- 1 motion motion 17084 Jan 6 18:40
72998 -rw-r--r-- 1 motion motion 16833 Jan 6 18:40
72987 -rw-r--r-- 1 motion motion 1665571 Jan 6 18:43
72997 -rw-r--r-- 1 motion motion 16497 Jan 6 18:40
72996 -rw-r--r-- 1 motion motion 16471 Jan 6 18:40
72995 -rw-r--r-- 1 motion motion 15426 Jan 6 18:40
72994 -rw-r--r-- 1 motion motion 14838 Jan 6 18:40
72993 -rw-r--r-- 1 motion motion 14571 Jan 6 18:40
72992 -rw-r--r-- 1 motion motion 15005 Jan 6 18:40
72991 -rw-r--r-- 1 motion motion 15866 Jan 6 18:40
72990 -rw-r--r-- 1 motion motion 15933 Jan 6 18:40
72988 -rw-r--r-- 1 motion motion 15760 Jan 6 18:40
73835 drwxr-xr-x 4 pi www-data 4096 Jan 6 19:35
73942 drwxrwxrwx 2 pi www-data 106496 Jan 6 20:49
root@raspberrypi: /var/www/images#
```

► Имена всех файлов в каталоге **images** содержат событие, дату и порядковый номер.

функции PHP, такие как `strstr()`, `preg_replace()`, `substr_replace()`, `substr()`, `date()` и `unlink()`.

Выделенная из имени файла дата с помощью простого оператора `if()` сравнивается с датой, отстающей на два дня от текущей. Это легко переделать — достаточно изменить 2 на требуемое количество дней от текущей даты. Если условие верно, файл удаляется функцией `unlink()`. Поскольку в этой системе используются только отдельные файлы, а не база данных, их довольно просто переместить в другое место, а так как это включает копирование и перемещение файлов, на ум приходят два способа. Один — с использованием такого пакета, как *rsync*, другой — обычное сжатие выбранных файлов в *ZIP* или *TAR* и копирование их на новое место командой `scp`. Простой пример `scp` показан ниже:

```
scp -P 22 /var/www/images.tar pi@example.com:/home/pi/images.tar
```

Вот и все. Вы создали собственную систему видеонаблюдения и записи обнаруженного движения, которую можете подправлять в соответствии со своими потребностями. Мы создали лишь голый скелет и файлы для просмотра видео, записи файлов и резервного копирования, и при желании всё можно улучшить. Простейшие подсказки — добавить подходящий шаблон в файлы **video.php** и **shadowbox.php** и приукрасить страницы с небольшой помощью волшебства CSS.

А сверх того можно настроить web-камеры и другие источники и сделать их доступными для просмотра общедоступно или друзьям — в зависимости от того, что вам нужно. Удачи! **LXF**

Скорая помощь

В имя файла копии можно включить время, чтобы файлы резервных копий не перезаписывались. В дополнение к этому, можно запустить задание *cron*, которое будет выполнять эту процедуру регулярно.

Nginx и Motion

Nginx в отличие от *Apache* не сразу готов для работы с *Motion*. На самом деле, после установки *Nginx* нужно предпринять ряд шагов, чтобы все заработало.

В случае с Raspberry Pi можно изменить значение параметра `worker_processes` с 4 на 1, в файле `/etc/nginx/nginx.conf`. Это рекомендуется сделать, так как у Pi всего одно ядро процессора.

Затем нужно изменить web-каталог по умолчанию на `/usr/share/nginx/www`. Для этого откройте файл `/etc/nginx/sites-enabled/sites-enabled/default`. Изменение показано далее — web-каталог изменен на `/var/www`:

```
#root /usr/share/nginx/www; root /var/www;
После предыдущего этапа можно быстро установить fastcgi, командой
apt-get install php5-fpm
Далее откройте файл /etc/nginx/sites-available/default и измените несколько строк кода, чтобы они были похожи на приведенные ниже. Обычно достаточно удалить несколько комментариев:
location ~ /\.php$ {
    fastcgi_split_path_info ^(.+\.php)(/.+)$;
    # NOTE: Надо, чтобы в php.ini было
    "cgi.fix_pathinfo = 0;"
    # Только с php5-cgi:
```

```
# fastcgi_pass 127.0.0.1:9000;
# С php5-fpm:
fastcgi_pass unix:/var/run/php5-fpm.sock;
fastcgi_index index.php;
include fastcgi_params;
}
```

Всё почти готово. Теперь откройте файл `/etc/php5/fpm/php.ini` и удалите еще один комментарий, чтобы получилась строка, похожая на такую:

```
cgi.fix_pathinfo=1
Наконец, после выполнения всех изменений перезапустите Nginx. Это делает команда
/etc/init.d/nginx restart
```

Ghost: Создайте свои темы

Стивен Ву рассказывает, с чего начать и как создать собственную тему для платформы блогов с открытым кодом Ghost.



Наш эксперт

Стивен Ву — веб-разработчик-фрилансер Magento и Wordpress. Подпишитесь на его ленту в Twitter, @designtodevelop, чтобы узнать о его проектах.



Раньше мы уже рассказывали об основах работы с Ghost [См. Учебники, **LXF183**, стр. 70], но на случай, если вы этого не читали, сообщаем: Ghost — это свободная платформа с открытым кодом для создания блогов. Она весьма успешно получила финансовую поддержку на Kickstarter в мае 2013, превзойдя изначально указанную сумму в £25000 и собрав более £196000. Созданный Джоном О'Ноланом [John O'Nolan], Ghost имеет единственную цель: предоставление блоггерам простого интерфейса, который позволяет им писать и публиковать свой контент, не отвлекаясь на сложности разработки, присущие традиционным платформам.

Ghost создан с нуля. Его простой и понятный интерфейс позволяет быстро просматривать архив, так что вы тратите меньше времени на управление своим блогом и больше времени на его создание. В нем имеется умный экран для ввода текста, использующий Markdown с просмотром в реальном времени на правом экране и простой функцией drag-and-drop для добавления изображений.

У Ghost три основных принципа: во-первых, он разработан для пользователей, а не для разработчиков, в отличие от большинства платформ создания блогов и CMS. Во-вторых, у платформы имеется лицензия MIT, так что вы можете делать с платформой всё,

что хотите, с некоторыми ограничениями. В-третьих, он создан во имя любви. Ghost — некоммерческая организация, а значит, ее цели — поддержка блоггеров, а не удовлетворение инвесторов. В нашем учебнике мы покажем, как установить и настроить Ghost локально и создать свою первую тему.

Чтобы приступить к созданию нашей темы Ghost, начните с папки установки Ghost [руководство по установке см. вверху стр. 81]. В `content/themes` создайте директорию новых тем с именем `mytheme`, или более интересным — только убедитесь, что оно написано маленькими буквами и без пробелов (допустимы дефисы). Это будет директория для размещения нашей кодовой базы. В этой директории создайте папки и файлы с такой иерархией:

assets/
css/
normalize.css
screen.css
images/
js/
fonts/
partials/
header.hbs
default.hbs
index.hbs
post.hbs

И `index.hbs`, и `post.hbs` — единственные файлы, необходимые для действующей темы. Без них вы получите ошибку.

Теперь в панели управления Ghost перейдите в Settings > General. В Theme выберите новую тему, которую вы только что создали под названием `mytheme`. Если ее нет, перейдите в терминал и перезапустите Ghost. Нажмите на Save, чтобы активировать эту тему. В интерфейсе вы пока что ничего не увидите. Это потому, что нам еще нужно добавить в нашу тему разметку.

Работа с Handlebars

Ghost использует язык шаблонов под названием Handlebars.js (<http://handlebarsjs.com>), и его предопределенные выражения облегчают создание и поддержку тем Ghost. Handlebars отделяет для вас шаблоны от чистого HTML. Помните, что в Handlebars вы не можете писать функций или фиксировать переменные. Handlebars разработан просто для отображения контента там, где выводятся выражения. Выражения Handlebars заключены в фигурные скобки и выглядят примерно так: `{{author.name}}`. Это соответствует свойству `author.name` и выводит имя автора.

Итак, давайте приступим к созданию нашей темы. Откройте файл `default.hbs` в своем любимом текстовом редакторе. Это базовый шаблон, и он включает все основные теги `<html>`, `<head>`, `<body>`, которые будут использоваться на вашем сайте, работающем на Ghost.

Скорая помощь

Все файлы, необходимые для этого урока, можно найти на GitHub Стивена по адресу <http://bit.ly/BuildAGhostTheme>.

Установка Ghost

Ghost — легковесное web-приложение JavaScript, созданное на Node.js. На его локальную установку, подготовку к работе и запуск требуются какие-то минуты. Вы можете установить Node.js из репозитивов вашего дистрибутива — в них почти наверняка есть версия из нужной последней стабильной серии 0.10.x. Пользователи Debian/Ubuntu/Mint найдут их с помощью

```
$ sudo apt-get install nodejs
```

Если вам нужна более поздняя версия, можете перейти на <http://nodejs.org> и скачать ее, или, если вы предпочитаете более аккуратную установку, познакомьтесь с инструкциями по установке

для разных дистрибутивов на <http://bit.ly/InstallingNodejs>. Сам Ghost в большинстве дистрибутивов в официальные пакеты не включен, поэтому перейдите на <https://ghost.org/download>, где есть свежая версия, и распакуйте ее. Для установки Ghost запустите в терминале

```
$ cd /path/to/downloads/ghost
```

```
$ npm install --production
```

Менеджер пакетов javascript, который идет с Node.js и называется *npm*, установит все требуемые зависимости в среду разработки и создания. По завершении этого процесса вы можете запустить Ghost в режиме разработки командой

```
$ npm start
```

В своем браузере перейдите в заново установленный блог Ghost на <http://127.0.0.1:2368>, и можете зарегистрировать свой логин администратора на <http://127.0.0.1:2368/ghost>. Зайдя на этот URL, вы увидите, что Ghost уведомляет вас об отправке сообщений электронной почты и использовании Ghost URL. Настройте эту функцию, изменив базовый URL. Откройте **config.js** в своем текстовом редакторе и измените параметры `url` в утверждениях **Development** или **Production**.

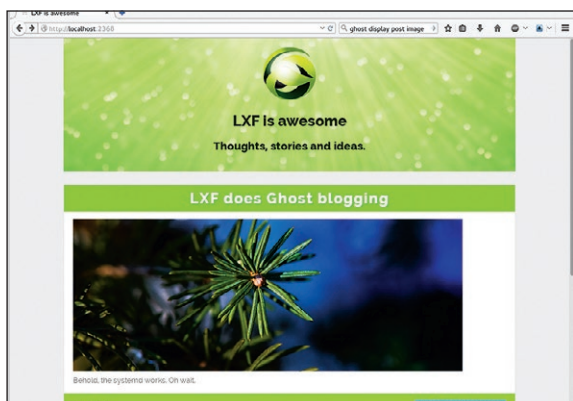
Закончив разработку, выйдите из системы, нажав в терминале `Ctrl+C`.

В этом шаблоне мы введем наш HTML doctype, основные мета-теги и теги заголовка и текстовой части. (См. **default.hbs** в файлах руководства на GitHub здесь: <http://bit.ly/BuildAGhostTheme>.) Вы заметите теги выражений: **{{! Responsive Meta Tags }}**. Все выражения в фигурных скобках, перед которыми стоит восклицательный знак, являются комментариями и не будут печататься в конечном исходном коде.

Вспомогательная функция `[helper] {{ghost_head}}` используется для вывода всех системных скриптов, стилей и мета-тегов. Вспомогательная функция `{{ghost_foot}}` используется для вывода скриптов внизу документа. Выражение Handlebars `{{body}}` имеет большое значение. Именно здесь будет отображаться весь ваш контент, который расширяет шаблон по умолчанию. Следующее за ним `{{body_class}}` используется, чтобы автоматически сгенерировать имена классов CSS для определенных страниц:

```
<body class="{{body_class}}">
  <div class="mytheme_page">
    {{{body}}}
  </div>
  {{ghost_foot}}
</body>
Index.hbs
```

Теперь в нашем **index.hbs** (см. исходный код в файлах проекта на GitHub) мы используем выражение Handlebars `{{!<default}}` в самом начале документа для ссылки на наш предыдущий базовый шаблон. Этот шаблон будет использоваться для нашей домашней страницы. Нам нужно выбрать стиль для каждого блога во вспомогательной функции `foreach`. Цикл открывает `{{#foreach posts}}`, а закрывает `{{/foreach}}`, и все, что внутри, будет отображать каждый пост с разметкой.



► Вы можете использовать помощник `{{date}}` для вывода опубликованной даты и управлять форматом даты через опцию `format`.

Для отображения контента каждого поста мы используем выражения Handlebars `{{content}}`. Мы также можем ограничить количество слов с помощью параметра `words="100"`.

Вы увидите, что все имена классов обрабатываются с помощью `mytheme_`. Это рекомендуемая практика при создании темы Ghost. Ghost автоматически присвоит определенные имена классов и ID определенным элементам вашей темы. Вам нужно будет избежать противоречий и продумать границы имен классов.

Partials

Обычно мы можем вставить нашу разметку заголовка прямо под `{{!<default}}`, однако преимуществом шаблонов Handlebars является иерархическая поддержка, посредством которой один шаблон может расширять другой. Сюда входит использование частичности. Это помогает устранить повторение кода и способствует возможности повторного использования. Мы можем выделить наш заголовок в частичный шаблон.

В директории **partials** откройте **header.hbs**. В панели управления Ghost Settings вы можете загрузить логотип вашего блога и его титульное изображение. Мы будем использовать выражение `if` для проверки наличия титульного изображения блога. Если оно существует, мы выведем его в качестве фонового изображения:

```
{{#if @blog.cover}}
  style="background-image: url('{{@blog.cover}}')"
```

```
{{/if}}
```

На сей раз проверим наличие логотипа блога.

```
{{#if @blog.logo}}
```

```
<a class="blog-logo" href="{{@blog.url}}">
```

```

```

```
</a>
```

```
{{/if}}
```

Средство доступа к глобальным данным `@blog` имеет доступ к глобальным настройкам в Ghost, которые мы можем вывести в нашей теме.

Теперь перейдем к занимательной части создания нашей темы. В нашей теме мы сделали ссылку на **normalize.css** для перенастройки готовых для HTML5 CSS. Именно в **screen.css** мы будем вводить все наши индивидуальные стили. Затем мы добавим глобальные стили, после чего — стиль нашего заголовка, и зададим **max-width**, чтобы наша структура не вышла за определенный размер в пикселях:

```
.mytheme_page {
  max-width: 980px;
  margin: 0 auto;
}
.mytheme_header {
  padding: 20px 0;
```

»

**Через месяц:
Идем в ногу
со временем
с NTP**

```

text-shadow: 2px 2px 2px rgba(26, 26, 26, 0.1);
text-align: center;
color: #2f3727;
}

Теперь создайте стиль каждого блога внутри контейнера его
статьи:

main article {
margin: 30px 0;
border-left: 1px solid #DBDBDB;
border-right: 1px solid #DBDBDB;
background-color: #FFFFFF;
}

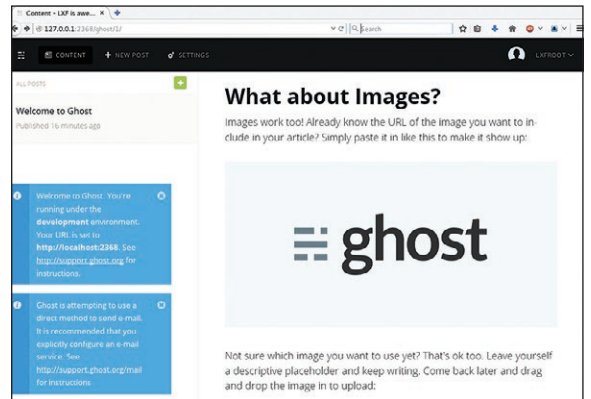
.mytheme_post_content {
padding: 0 20px;
}

.mytheme_post_title {
margin: 0 0 20px 0;
padding: 10px;
font-size: 2em;
letter-spacing: 2px;
text-align: center;
text-shadow: 2px 2px 2px rgba(26, 26, 26, 0.2);
color: #FFFFFF;
background-color: #8ACD36;
}

.mytheme_post_title a {
text-decoration: none;
color: #FFFFFF;
}

.mytheme_main_img img {
width: 100%;
max-width: 100%;
border: 0;
}
    
```

Разместите дату слева, а напротив — кнопку Read More [Читайте дальше]. Создайте этой ссылке презентацию кнопки:



➤ После установки Ghost вы увидите URL разработки и предупреждение о настройке сервиса электронной почты.

```

.mytheme_post_info {
overflow: auto;
padding: 0 20px;
background-color: #98C148;
}

.mytheme_date {
float: left;
padding-top: 20px;
color: #FFFFFF;
}

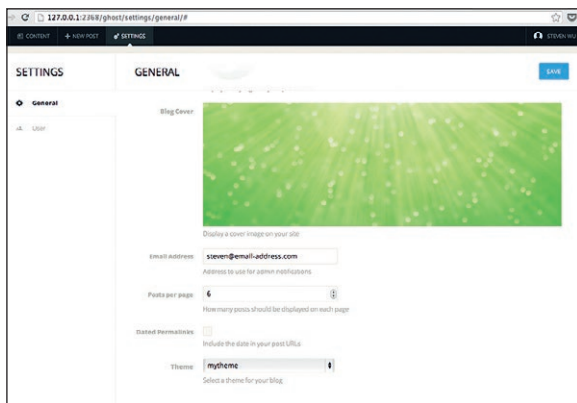
.button {
float: right;
padding: 20px 0;
}

.button a {
padding: 5px;
transition: ease .3s;
text-decoration: none;
color: #FFFFFF;
background-color: #39A9DA;
}

.button a:hover {
background-color: #199ED9;
}
    
```

Мы коснулись основных аспектов разработки темы Ghost. Надеемся, эта информация откроет для вас возможности создания своей персональной темы. Вы увидите, что при использовании предварительно определенных выражений Handlebars создание темы Ghost — процесс очень простой. Не забудьте скачать файлы, сопровождающие наш урок, с GitHub по: <http://bit.ly/BuildAGhostTheme>. Также загляните в документацию, чтобы найти более подробную информацию, которая поможет вам создавать и настраивать собственную тему, на <http://docs.ghost.org/themes>. **LXF**

➤ Выберите вновь созданную тему в панели управления, чтобы ее активировать. Возможно, для выбора темы придется перезапустить Ghost в терминале.



Подготовка post.hbs

После настройки всего лишь шаблона домашней страницы, **post.hbs** будет контролировать отображение всех страниц блогов. Опять же в том же шаблоне (см. файл **post.hbs** в папке проекта, которую можно скачать с GitHub здесь: <http://bit.ly/BuildAGhostTheme>), мы используем те же выражения Handlebars: **{{!<default}}** и **{{> header}}**. На этот раз мы используем открывающие выражения **{{#post}}** и **{{!post}}** для отображения одного поста блога.

И в контексте одного поста мы имеем доступ к данным об авторе, в том числе, его имя и биографические данные.

Мы можем отобразить информацию об авторе, просто добавив приведенный ниже код:

```

<section class="author">
<h3>Written By:</h3>
<h4>{{author.name}}</h4>
<p>{{author.bio}}</p>
</section>
    
```

</section>

Теперь мы можем применить к нашему одиночному посту блога стили CSS. Рекомендуется размещать все ваши стили для каждого поста блога в отдельном файле CSS. Это важно, потому что в ближайшем будущем Ghost продемонстрирует новую функцию разделения экрана, которая будет загружать все стили индивидуальных тем в интерфейсе администратора.

1С:ПРЕДПРИЯТИЕ 8

Новое поколение систем автоматизации управления и учета

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

Теперь работает в Linux!

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



Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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



1С: Не только для Windows

Михаил Вознесенский проявил заботу о малых и больших предприятиях, которые были вынуждены маяться с ОС от Microsoft из-за нормативных требований.



Наш эксперт

Михаил Вознесенский — добропорядочный пользователь операционных систем семейства GNU/Linux, стремящийся показать, что работать с ними просто и свободно.

Н и для кого не секрет, что в России 1С является монополистом в сфере программных продуктов для финансовых расчётов, предоставления документов в налоговую и т. д. Но не многие знают, что с выходом платформы 8.3 появился полноценный клиент под семейство операционных систем GNU/Linux. На нашем уроке я расскажу о том, как можно и нужно устанавливать 1С и какие подводные камни встречаются на этом пути.

Из чего состоит 1С

Когда речь заходит о версии 1С, многие люди путают версию конфигурации и версию платформы. Так вот, конфигурация и платформа — совершенно разные вещи, причем конфигурация косвенно зависит от платформы. Платформа — это специальная программа от 1С, которая позволяет запускать созданные или перенесённые базы 1С в режиме Предприятия, для работы или в режиме Конфигуратор, для программистов. Конфигурация — это набор уже созданных правил, обработок, документов и т. д., на основе которых и работают люди, создавая и заполняя свои базы данных. К конфигурациям как раз-таки относятся бухгалтерия, управление торговлей и т. д. То есть при покупке какой-либо комплектации 1С вы получаете конфигурацию, на основе которой работаете и предоставляете отчётность, и платформу, благодаря которой осуществляется запуск вашей конфигурации. Поэтому версия конфигурации и версия платформы — это совсем разные вещи. Косвенная связь конфигурации и платформы выражается в том, что если конфигурация достаточно новая, а платформа старая, то 1С может запросить обновление платформы.

Подготовительные работы

Прежде чем приступать к установке 1С на компьютер, надо провести подготовительные работы. Они заключаются в установке вспомогательных пакетов (с их списком более подробно можно ознакомиться в справке 1С — Особенности работы в Linux).

Первое, что понадобится — установить шрифты из состава Microsoft Core Fonts. В таких дистрибутивах, как Debian, Ubuntu или Mint, пакет этих шрифтов содержится в репозиториях и имеет название `ttf-mscorefonts-installer`.

У RPM-дистрибутивов дела обстоят немного по-другому. Сама 1С предлагает пройти по ссылке <http://corefonts.sourceforge.net/> и установить по инструкции, в противном случае — скачать всё вручную с sourceforge.net (<http://sourceforge.net/projects/corefonts/files/the%20fonts/final/>) и расположить в директории пользователя (`$HOME/fonts`), откуда и будет запускаться 1С. Я вам предлагаю более простой путь. Во-первых, проверьте свои репозитории на наличие данного пакета. В дистрибутиве ROSA данный пакет распространяется под именем `webcore-fonts`, в других дистрибутивах может быть иное название. Если всё-таки в ваших репозиториях его нет, то возможен вариант скачать пакет `msttcorefonts` с сайта rpm.pbone.net (<http://rpm.pbone.net/index.php3/stat/4/idpl/25430132/dir/other/com/msttcorefonts-2.5-1.noarch.rpm.html>).

Следующее, что нам предстоит сделать — установить пакет `ImageMagick`. Без этого пакета 1С наотрез откажется запускаться и будет выдавать ошибку вида «Не обнаружена библиотека `libWand.so`». Иногда после установки данного пакета 1С всё равно способна «ругаться» на отсутствие данной библиотеки. Для устранения этой ошибки придётся указать системе на библиотеку вручную, через посредство символической ссылки. Поищем нашу библиотеку:

```
sudo find /usr/lib64/ -name "libMagickWand*" -print
```

Результатом будет нечто вроде такого:

```
/usr/lib64/libMagickWand-6.Q16.so.2.0.0
```

```
/usr/lib64/libMagickWand-6.Q16.so.2
```

Это для 64-битной системы; для 32-битной будет каталог `/usr/lib/`.

Итак, путь к библиотеке мы разобрали; создадим же символическую ссылку для корректной работы 1С.

```
ln -s -T /usr/lib64/libMagickWand-6.Q16.so.2 /usr/lib64/libWand.so
```

Последнее, что остается — это установить пару пакетов, имена которых содержат следующее: `FreeType` и `unixODBC`. Ещё раз напоминаю: полный перечень внешних пакетов указан в справочнике 1С — Особенности работы в Linux; для обычной работы больше ничего не понадобится.

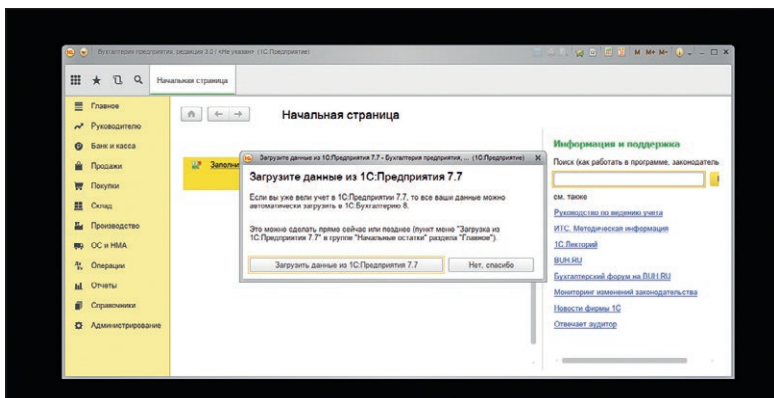
Важно отметить: если ваша система 64-разрядная, то и 1С надо устанавливать 64-разрядную. Это необходимо для того, чтобы избежать мелких конфликтов с системой.

Установка

Разобравшись, из чего состоит 1С, и проведя подготовительные работы, приступим к установке.

Платформа распространяется в двух вариантах — в виде Deb- и RPM-пакетов, что означает установку 1С в системах соответственно на базе Debian (Ubuntu) либо Fedora.

» Работаящая 1С на ОС ROSA.



Полный перечень пакетов таков:

- » **1C_Enterprise-client** Толстый клиент 1С: Предприятие 8;
- » **1C_Enterprise-thin** Тонкий клиент 1С: Предприятие 8;
- » **1C_Enterprise-common** Общие компоненты 1С: Предприятие 8;
- » **1C_Enterprise-server** Компоненты сервера 1С: Предприятие 8;
- » **1C_Enterprise-ws** Адаптер для публикации web-сервисов 1С: Предприятие 8 на web-сервере на основе Apache HTTP Server 2.0 или Apache HTTP Server 2.2;
- » **1C-Enterprise-crs** Компоненты сервера хранилища конфигурации 1С: Предприятие 8 (только для платформы i386)

Пакеты-двойники, содержащие в названии суффикс “-nls” — это дополнительные национальные ресурсы для соответствующего им пакета.

Теоретически 1С можно установить на Arch Linux, что потребует немало трудов и известного понимания, но на просторах Интернета можно отыскать умельцев, которые изловчились это сделать — они охотно вам помогут.

Сам же установщик конфигурации распространяется в одном файле (**setup**), являющимся скриптом оболочки UNIX, что не мешает его установить на любой дистрибутив GNU/Linux.

На сайте 1С есть перечень рекомендуемых ОС (<http://v8.1c.ru/requirements/>). Стоит отметить тот факт, что для Тонкого или Толстого клиента указаны весьма старые дистрибутивы Linux, а для остальных вариантов 1С указывается самая минимальная версия дистрибутива и выше.

Несмотря на то, что установочные пакеты платформы имеют популярное расширение, установку рекомендуется проводить строго через командную строку.

Установка на домашний компьютер

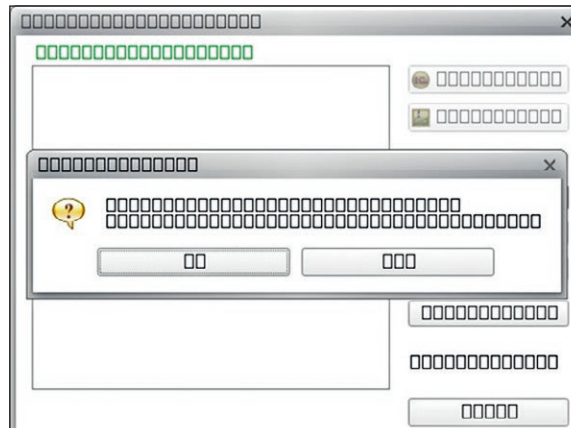
Пожалуй, самая простая в отладке, так как всё действие происходит непосредственно на вашем личном компьютере. Раз у нас домашний компьютер, понадобится установка только клиента. Для его инсталляции потребуются следующие пакеты:

- » *common* » *server* » *client*
- » *common-nls* » *server-nls* » *client-nls*

Вы, возможно, озадачитесь, увидев в списке пакеты *server* и *common*, поскольку если скачать архив установщика клиента для Linux с сайта <http://users.v8.1c.ru/>, то в архиве будет только два пакета: *client* и *client-nls*. Эти четыре добавочных пакета нужны для устранения зависимостей, так как если устанавливать просто два пакета *client*, то... возможны варианты — пакет установится, но будет работать некорректно, или не установится и скажет, что для удовлетворения зависимостей необходим пакет *server*. Поэтому для нормальной работы лучше поставить все шесть пакетов.

Пакеты можно установить поочередно, именно в том порядке, в котором они перечислялись (это нужно для установки правильных зависимостей пакетов), или поместить их в одну папку и установить одной командой. Здесь следует отметить выбор пакетного менеджера для установки. Я рекомендую сначала попробовать обычный пакетный менеджер, который вы используете для установки повседневных программ — такой как *apt-get* в семействе Debian, *yum* в семействе Red Hat, *urpm* в семействе Mandriva. Если они не сработали, воспользуйтесь стандартными менеджерами, которые по умолчанию обычно всегда присутствуют в системе. Это *dpkg* из семейства Debian и *rpm* из семейства Red Hat и Mandriva. Для чего это нужно? Для автоматической установки необходимых библиотек: в противном случае 1С не установится и выведет на экран список недостающих библиотек. Поэтому лучше всего при установке пользоваться стандартными пакетными менеджерами для построения правильных зависимостей.

Далее можно перейти к установке конфигурации (шаблона). Файл, который нам необходим, имеет имя **setup**; не перепутайте его с **setup.exe**! 1С распространяет инсталлятор конфигурации (или обновления) вместе с инсталлятором для Windows,



» Некорректно отображавшиеся шрифты на ROSA до установки 32-битного пакета *rango* или переустановлении 64-битной версии 1С.

и в одном каталоге лежат установщик для Windows (нам он, естественно, не нужен) и для GNU/Linux.

Что нам нужно будет сделать?

- 1 Перейти в каталог шаблона: **cd /home/ваш_логин/каталог_инсталлятора**.
- 2 Сделать файл **setup** исполняемым: **chmod +x setup**.
- 3 Запустить его: **./setup**.

После проделанных манипуляций перед вами откроется графическое окно, в котором вы указываете директорию и устанавливаете. Вот и вся установка конфигурации.

Теперь переходим к кульминационному моменту. Это сам запуск 1С. Если все описанные действия были выполнены корректно, то 1С стартует — можете создавать базу и начинать работать. А если возникла какая-то ошибка или вылет программы, тогда запустим 1С через консоль, следующей командой:

```
/opt/1C/v8.3/x86_64/1cv8
```

(это для 64-битной системы, для 32-х вместо **x86_64** будет **i386**). В консоли будут отображены все ошибки.

К примеру, в 64-битном дистрибутиве ROSA после установки 32-битной 1С шрифты отображались некорректно: вместо букв появлялись квадраты. Выяснилось, через консоль, что отсутствовал 32-битный пакет *rango*. Было два варианта решения данной проблемы:

- » установка 32-битного пакета *rango*;
- » установка вместо 32-битной, 64-битной 1С.

Установка на выделенный компьютер

Иногда возникает необходимость публикации базы через web-интерфейс, чтобы работать с 1С не только дома или в офисе, но и удалённо. Для таких случаев в 1С есть программа, которая позволяет публиковать вашу базу в сети, после чего появляется возможность доступа к базе через web-браузер или тонкий клиент.

Помимо Linux-машины, понадобятся *Apache* (версии 2.0 или 2.2) и внешний IP-адрес (для доступа к 1С извне локальной сети).

Для установки 1С необходимы следующие пакеты:

- » *common* » *server* » *ws*
- » *common-nls* » *server-nls* » *ws-nls*

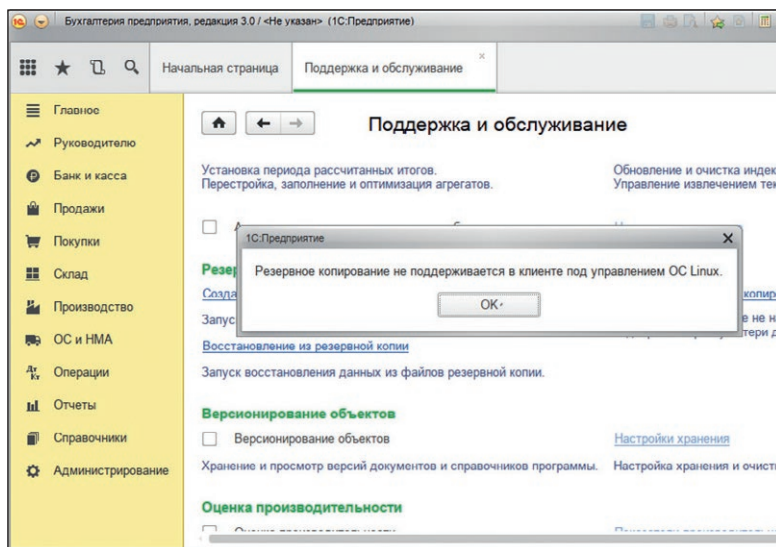
После установки нужно создать каталог в директории *Apache*, где будет размещаться конфигурационный файл нашей публикации — **default.vrd**, и выбрать/создать директорию, где будет храниться база. К примеру:

» **/var/www/html/1c** — директория конфигурационного файла публикации.

» **/1C/Accounting** — каталог базы. После его создания скопируйте туда существующую базу, но не всю: достаточно одного файла с расширением .CD (по умолчанию **1cv8.CD**).

А теперь сама публикация. Здесь поможет утилита от 1С — *webinst*, располагающаяся в каталоге, где установлена сама 1С: **/opt/1C/v8.3/x86_64/webinst**. При обычном запуске утилита

»



➤ Для Linux резервное копирование ИБ пока не работает.

выведет инструкцию команд с возможными параметрами для публикации базы, удалению публикации и т.д.

Вот пример команды публикации:

```
./webinst -apache22 -wsdir 1c -dir /var/www/html/1c/ -connStr 'File= "/1C/Accounting"' -confPath /etc/httpd/conf/httpd.conf
```

Ее параметрами являются:

- **-apache22** — версия используемого *Apache 2.2*.
- **-wsdir** — имя алиаса для соединения с базой (http://ваш_ip/1c/).
- **-dir** — путь к каталогу, где будет располагаться конфигурационный файл публикации **default.vrd**.
- **-connStr** — путь к каталогу, где располагается информационная база. В данном примере 1С установлена в файловом варианте; если установлен сервер 1С: *Предприятие*, то путь будет выглядеть приблизительно так: **Srvr=1c8;Ref=Accounting**. Через установленный клиент 1С можно узнать расположение базы: данная информация располагается внизу приложения приветственного окна перед выбором запуска какой-либо базы.
- **-confPath** — путь конфигурационного файла *Apache* (при публикации параметр необязательный).

Так как Linux обладает механизмом контроля прав доступа, необходимо каталогам (**/var/www/html/1c**, **/1C/Accounting**) и вложенным в них файлам присвоить пользователя и группу apache. Плюс дополнительно надо расширить данный каталог информационной базы для пользователя **usr1cv8** (создаётся автоматически) и пользователя, от имени которого будет происходить сама работа с базой. Выполняется это двумя командами:

```
setfacl -m u:usr1cv8:rwx -R /1C/Accounting
setfacl -m d:u:usr1cv8:rwx -R /1C/Accounting
```

где **usr1cv8** — это пользователь, а **/1C/Accounting** — каталог, где хранится база 1С.

На этом публикация выполнена. Перезапускаем *Apache*:

```
/etc/init.d/httpd restart
```

Это команда для CentOS (в других дистрибутивах она может отличаться).

Чтобы проверить публикацию, введите в URL-строку браузера следующий адрес: **http://ip_выделенного_компьютера/1c**. Если всё в порядке, 1С начнёт загружаться.

Для запуска информационной базы через платформу выполните следующие действия:

- 1 Щелкнуть по кнопке «Добавить».
- 2 Из списка выбрать добавление уже существующей базы.
- 3 Выбрать пункт расположение на web-сервере.
- 2 После этого вписать ваш URL-адрес.

Также возможен вариант настроить базу по определённому порту, в том числе и по 443.

Установка на сервер

Чтобы установить сервер 1С:Предприятие, нам понадобится сама 1С и PostgreSQL. Обычная PostgreSQL, которая есть в репозиториях, здесь не подойдёт: нужна специальная, «пропатченная» для 1С. Есть несколько способов её установки:

- Установить PostgreSQL от Etersoft.
- Установить PostgreSQL от 1С (скачав с сайта <http://users.v8.1c.ru/>).
- Установить PostgreSQL от 1С с изменёнными пакетами.

Первые два способа установки тривиальны, поэтому я ограничусь описанием третьего.

Следует отметить ещё тот факт, что инсталляторы СУБД от 1С различаются набором входящих пакетов для случаев Deb- и RPM-дистрибутивов. Для RPM-дистрибутивов, как мне известно, с установкой PostgreSQL от 1С нет никаких проблем, чего о Deb-дистрибутивах не скажешь. Поэтому описанная ниже инструкция по сборке установочных пакетов PostgreSQL будет ориентирована на Deb-дистрибутивы. Для RPM-систем достаточно установить без изменений инсталлятор от 1С.

Но сначала подготовим систему. Изменим размер сегмента памяти: откроем для редактирования конфигурационный файл (можно использовать любой редактор; в данном случае используется консольный редактор *nano*) —

```
nano /etc/sysctl.conf
```

и добавим следующие две строки в конец файла: строку

```
kernel.shmall=524288
```

— этот параметр высчитывается путём деления объёма оперативной памяти в байтах на 4096 — и строку

```
kernel.shmmax=1073741824
```

— этот параметр задаёт максимальный объём 1 Гб: допускается использовать до 50% оперативной памяти, а в нашем случае имеется всего 2 Гб ОЗУ.

Сохраняем вписанные данные и выходим. Применяем изменённые параметры:

```
sysctl -p
```

Теперь генерируем русскую локаль:

```
locale-gen en_US ru_RU ru_RU.UTF-8
```

```
export LANG="ru_RU.UTF-8"
```

Далее собираем установщик PostgreSQL. В качестве примера будет показана версия 9.2.4 для x86_64 Ubuntu 14.04.

Скачав инсталлятор СУБД с сайта 1С, оставим только следующие пакеты:

- **libpq5_9.2.4-1.1C_amd64.deb**
- **postgresql-client-9.2_9.2.4-1.1C_amd64.deb**
- **postgresql-contrib-9.2_9.2.4-1.1C_amd64.deb**
- **postgresql-9.2_9.2.4-1.1C_amd64.deb**

Прочие пакеты —

- **postgresql-client-common_140~lucid_all.deb**
- **postgresql-common_140~lucid_all.deb**

заменяем на

- **postgresql-client-common_154_all.deb**
- **postgresql-common_154_all.deb**

Перечисленные пакеты можно скачать с зеркала Яндекса (<http://mirror.yandex.ru/ubuntu/pool/main/p/postgresql-common/>).

Остался заключительный этап сборки инсталлятора. Необходимо пересобрать пакет **postgresql-contrib-9.2_9.2.4-1.1C_amd64.deb**. На это уйдут четыре команды:

```
dpkg -x postgresql-contrib-9.2_9.2.4-1.1C_amd64.deb tempost
```

— распаковываем пакет в созданный каталог **tempost**;

```
dpkg -e postgresql-contrib-9.2_9.2.4-1.1C_amd64.deb tempost/DEBIAN
```

— ещё раз дораспаковываем пакет в созданный подкаталог **DEBIAN**;

```
nano /tempost/DEBIAN/control
```

— открываем в редакторе файл **control** и меняем "libc6" на "libc52";


```
dpkg -b tempost postgresql-contrib-9.2_9.2.4-1.1C_amd64_new.deb
```

— собираем данный пакет снова, уже с внесёнными изменениями.

Пакет *libicu52* устанавливаем из репозитория. Собственно, ради него мы и пересобирали пакет. В сборках СУБД от Etersoft и 1С используется библиотека *libicu46*, и иногда из-за неразрешённых зависимостей возникают проблемы в установке СУБД или обновлениях системы. Так как у нас Ubuntu 14.04, мы заменили библиотеку *libicu46* на *libicu52*.

Затем необходимо установить СУБД. После сбора всех необходимых пакетов, наш инсталлятор будет содержать следующее:

- » *libpq5_9.2.4-1.1C_amd64.deb*
- » *postgresql-client-common_154_all.deb*
- » *postgresql-client-9.2_9.2.4-1.1C_amd64.deb*
- » *postgresql-common_154_all.deb*
- » *postgresql-contrib-9.2_9.2.4-1.1C_amd64_new.deb*
- » *postgresql-9.2_9.2.4-1.1C_amd64.deb*

Установка этих пакетов производится точно так же, как и установка пакетов самой 1С.

Проверяем *PostgreSQL*:

```
service postgresql status
```

Это команда для Ubuntu.

Если всё хорошо установилось, то будет выведено нечто наподобие

```
port 5432: online
```

Теперь для успешной работы с СУБД нам осталось задать пароль для суперпользователя *postgres* и поправить конфигурационный файл:

```
su postgres
```

— отныне мы выполняем команды от имени суперпользователя СУБД.

```
psql -U postgres -c "ALTER USER postgres WITH PASSWORD 'ваш_пароль';"
```

Далее для корректной работы 1С с базой данных надо подправить файл *pg_hba.conf*. По умолчанию он помещается в каталоге, где будут храниться сами БД. Откроем файл на редактирование:

```
nano /usr/lib/postgresql/9.2/bin/initdb/pg_hba.conf
```

— в нём нужно найти строки, подобные

```
host all all 127.0.0.1 ident
```

и во всех таких строках заменить “*ident*” на “*md5*”.

После этого сохраняем файл и перезагружаем СУБД:

```
service postgresql restart
```

Если всё хорошо, то на экране появится статус *DONE*.

На данном этапе установку *PostgreSQL* можно считать законченной. Осталось установить самоё 1С. Установка выполняется точно так же, как установка на домашний компьютер.

Перезапускаем сервер 1С:

```
service srv1cv83 restart
```

Ну вот, установка сервера 1С: *Предприятие* завершена.

Стоит отметить ещё один момент. Иногда при создании новой информационной базы выскакивает ошибка:

```
тип "nvarchar" не существует at character 31
```

Для решения данной проблемы необходимо пользователя СУБД временно «повысить» до суперпользователя с правом создания БД, даже в случае её отсутствия. Тогда информационная база будет успешно создана.

Лицензионная защита

В предыдущих пунктах я рассказал, как правильно устанавливать 1С и какие подводные камни могут встречаться, но ничего не говорил о лицензии. Поэтому сейчас опишу, какие виды лицензий бывают и как их нужно активировать.

Лицензии делятся на две категории: программные и аппаратные. У обеих категорий есть свои плюсы и минусы. Начнём с программной, так как она наиболее популярна.

Плюс программной лицензии в том, что она дешевле, минус — она привязывается к определённому компьютеру (на котором была активирована). Программная лицензия представляет собой ключ, или, точнее, PIN-код (специальный набор цифр). Активация этого ключа обеспечивает доступ к работе с программой.

Как же его активировать? Активация происходит непосредственно через графический интерфейс. При первом запуске информационной базы 1С на ПК выдаёт сообщение о том, что лицензия не обнаружена, и предлагает активировать её. Здесь проблем нет. Просто следуйте шагам активации.

В случае выделенного компьютера, на котором, к примеру, публикуется база, при запуске её через *web*-браузер тоже выведется сообщение о том, что лицензия не обнаружена, но активировать её не удастся. Придётся активировать её непосредственно на самом компьютере, где установлена 1С. Если у нас *Linux*-сервер, необходимо будет поднять графическую оболочку и доустановить пакет клиента. Только после этого можно будет активировать лицензию, но не спешите.

Помимо индивидуальной программной лицензии, существуют лицензии на несколько рабочих мест. В таких случаях обычно имеется два варианта ключей:

- » **Однопользовательские** — этот вид включает ровно столько ключей, на сколько рабочих мест была куплена лицензия, плюс два резервных (о резервных ключах немного позже).
- » **Многопользовательские** — один ключ плюс 2 резервных.

В чём их отличие? Отличие в том, что однопользовательские ключи ставятся на каждый компьютер через платформу ручную, а многопользовательские ставятся на один компьютер (как правило, это сервер, который и будет отслеживать количество подключённых пользователей). В случае *Linux* лучше всего устанавливать однопользовательские ключи, так как многопользовательские работают некорректно.

Ну, а что же резервные ключи? Они нужны для перелицензирования или переактивации 1С, в том случае, если компьютер был заменён на другой или была переустановлена ОС.

Теперь перейдём к аппаратным ключам. Аппаратный ключ представляет собой нечто вроде маленькой *USB*-флэшки. Здесь в некотором роде всё намного проще. Аппаратный ключ не привязывается к определённому компьютеру. Поэтому, к примеру, в случае замены компьютера вам не понадобится возиться ни с каким перелицензированием.

Установка на *Linux* тоже не составляет труда, плюс ко всему установка производится через терминал, поэтому графическую оболочку поднимать ни к чему. Для активации *USB*-ключа нам понадобятся драйверы *HASP*. Скачать их можно с сайтов компаний *SafeNet* и *Etersoft*. Как и в случае программных ключей, существуют ключи на одно и несколько рабочих мест. Для индивидуального ключа, на одно рабочее место, скачивается драйвер *HASP HL*, а для сервера — *Sentinel HASP*. На сайте *SafeNet* данные драйверы распространяются в *Deb*- и *RPM*-пакетах и ещё в скриптах оболочки *UNIX*.

На случай проблем с активацией лицензий предусмотрены контактный номер и электронная почта центра лицензирования программы 1С.

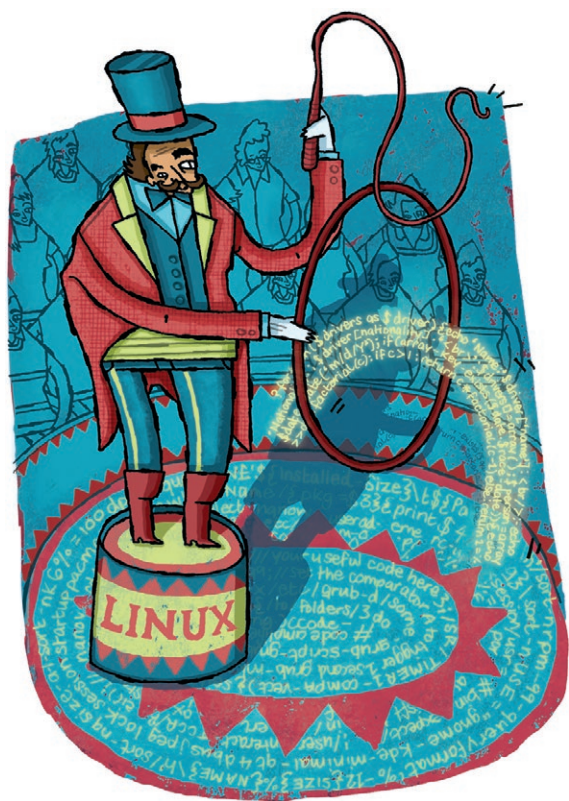
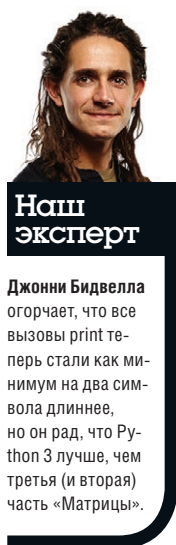
Заключение

Итак, на нашем уроке вы узнали, из чего состоит 1С, как правильно устанавливать 1С на *Linux*, какие бывают виды лицензий и как лицензии активировать.

Очень положительно то, что 1С затеяла поворот разработки в сторону *Linux*. Надеемся, с течением времени процесс установки будет более дружелюбен к пользователю ПК и будут реализованы все функции. На момент написания статьи замечено, что автоматическое резервное копирование информационной базы для семейства ОС *GNU/Linux* недоступно. **LXF**

Python: Нырнем в Версию 3

Джонни Бидвелл исследует один из, пожалуй, наименее любимых и уважаемых сиквелов за всю историю языков программирования.



Python 3.0 (а также Py3k или Python 3000) вышел еще в декабре 2008 года, но прошло семь лет, и многие до сих пор им не пользуются. В целом это происходит не потому, что программисты Python и разработчики дистрибутивов представляют собой шайку лодырей, да и ситуация очень отличается, например, от той, когда пользователям не удается обновить или они отказываются обновить (уничтожить?) на своих компьютерах Windows XP. Прежде всего, Python 2.7, хотя и явно последний в линейке 2.x, все еще регулярно поддерживается и, наверное, будет поддерживаться до 2020 года. К тому же, поскольку многие крупные проекты на Python (а также немало проектов помельче) так и не адаптированы для третьей версии, все, кто ими пользуются, привязаны к версии 2.7.

Ранее несколько крупных проектов — NumPy и Django — выполнили этот переход, и была надежда, что другие проекты последуют за ними, что приведет к лавинному эффекту. К сожалению, этого не произошло, и большая часть кода, которую вы найдете, не будет работать с Python 3. За несколькими исключениями, Python 2.7 обладает прямой совместимостью с 3.x, поэтому во многих случаях можно создать код, который будет работать в обеих версиях, но программисты все еще предпочитают делать

по-старому. Действительно, даже в этом уважаемом издании авторы, по привычке ли, по незнанию или по любви к прошлому, продолжают писать код, который никоим образом не совместим с Python 3. В данной статье такого не будет. Обещаем.

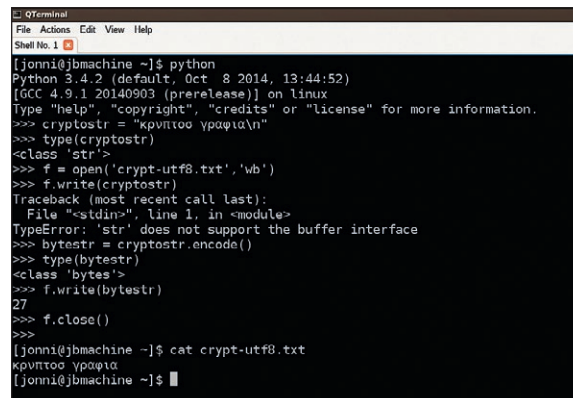
Примерно так могла бы выглядеть ваша программа-первенец на Python:

```
print 'Hello world'
```

И знаете что? Она не работает в Python 3 (а вы же только что обещали!). Причина в том, что `print` — это в Python 2 выражение, а в Python 3 — функция, а все функции без исключения вызываются с аргументами в скобках. Помните, что функции не обязаны что-то возвращать (такие функции называются `void` [пустые]), и `print` теперь — это «пустая» функция, которая, в простейшей форме, получает строку в качестве входного аргумента, выводит ее в стандартный поток вывода и не возвращает ничего. В некотором смысле `print` является функцией и в Python 2, так как ее можно вызвать со скобками; но было принято решение сделать для нее особый синтаксис и сокращенный вариант без скобок. Кстати, это очень похоже на признание, которого добился некий математик, когда его изобретение больше не пишется с заглавной буквы: например, «абелевы группы». Но подобные исключения не являются частью канона Python («Особые случаи недостаточно особые, чтобы нарушать правила»), поэтому функции всегда пишутся со скобками. На более глубоком уровне, `print` в виде функции обладает большей гибкостью для программистов — как встроенную функцию ее можно заменить, что может быть удобно, если вы боретесь с условностями или пишете функцию-обертку для поиска и игнорирования символов Unicode. Короче, ваша первая программа на Python должна была выглядеть так:

```
print ('Hello world')
```

И она полностью совместима с Python 2 и 3. Если вы любите запятую в конце выражений `print` (чтобы подавить переход на новую



► Греческая *Kryptos graphia*, что означает «тайнопись» [см. «Криптография старая и новая», LXF189, стр. 56], в правильном скрипте сопровождается переходом на новую строку.



Революция Unicode

До появления Unicode текст кодировался в ASCII. Каждому символу отвечал 7-битный код, что позволяло закодировать 128 символов. Часть кодов использовалась для непечатаемых символов телеайпа (ASCII появился в 1960-х), и если добавить к ним применяемые алфавитно-цифровые символы, оставалось не так уж много. Так как коды было удобно представить в виде целого байта, появилось несколько 256-символьных расширений таблицы ASCII. Самое известное из них — ISO-8859-1, иногда называемое Latin-1. Этот широко используемый набор символов (и сопутствующая кодировка Windows-1252) содержит почти все символы с диакритическими знаками, используемые в языках

на основе латиницы, а также символы, используемые в романизации других языков. В результате он довольно-таки распространен в Западном полушарии, но отнюдь не решает проблем в других частях света.

Правильным решением проблемы была бы стандартная кодировка (или несколько кодировок), которая охватывает максимально возможный набор символов, которые могут понадобиться каждому жителю Земли. Разумеется, это гораздо больше, чем 256 символов, поэтому нам придется распрощаться с кодировкой символа в один байт (из-за несоответствия кодов и байтовых кодировок), но оно и к лучшему. К счастью, все споры,

занесение в таблицы и другие сложные процедуры завершились, и у нас есть ответ: Unicode. Он охватывает более 100 000 символов, запись в двух направлениях, лигатурные формы и др. На данный момент используется две кодировки: UTF-8, где для «основных» символов используется 1 байт (благодаря чему UTF-8 полностью обратно-совместим с ASCII) и до четырех байт для более космополитичных; и UTF-16, где используется два байта для части символов и по 4 для остальных. Unicode был широко принят за стандарт, как для хранения данных, так и для тестов внутренней обработки. Основной смысл существования Python 3 в том, что его предшественник не умел делать последнего.

строку), то у меня для вас плохая новость: это больше не работает. Вместо нее используется параметр `end`, который по умолчанию означает переход на новую строку. Например, функции `print('All on', end=" ")` и `print('one line')` делают именно это.

Print в Python 3

Львиную долю программ на Python можно было бы сделать более совместимыми с третьей версией, просто изменив синтаксис `print`, но есть и масса других, менее тривиальных вещей, которые могут пойти не так. Чтобы понять их, сначала нужно узнать, что на самом деле изменилось в Python 3.

Большая часть мира не говорит по-английски и даже не пользуется латинским алфавитом, а те, кто все-таки им пользуется, часто снабжают символы различными видами ударений или диакритических знаков. В результате наряду со старым добрым стандартом ASCII появилось несколько других, несовместимых друг с другом. Каждой графеме (обобщенное название символа) назначается код, а каждому коду назначается байтовый код; иногда они совпадают. Раньше, если вы хотели поделиться документом с буквами иностранного алфавита в нем, чистый ASCII не помог бы — пришлось бы воспользоваться одной из альтернативных кодировок, при условии, что у тех, с кем вы делитесь, применяется такая же; но в общем приходилось обращаться к текстовому процессору с конкретным шрифтом, то есть перекидывать проблему куда-то еще. К счастью, сейчас у нас есть общепринятый стандарт: Unicode (см. врезку «Революция Unicode» вверху), который охватывает все базовые наборы символов и обратно-совместим с ASCII (в части кодов) и его расширением Latin-1. Символы Unicode можно даже употреблять в именах доменов, хотя внутри все они хранятся как ASCII, благодаря системе под названием Punycode.

Python 2 не имеет поддержки Unicode, но она довольно искусственная (строки Unicode просто перекодируются внутри), и некоторые сторонние модули работают с ней плохо. Строки в Python 2 могут быть типа `str` (который прекрасно поддерживает ASCII, но с кодами больше 127 ведет себя непредсказуемо) или `unicode`. Строки типа `str` хранятся в байтах и при выводе в терминал преобразуются в кодировку системной локали (через переменные окружения `LANG` и `LC_*` в Linux). В любом современном дистрибутиве ею скорее всего будет UTF-8, но это одна из тех вещей, в которых нельзя быть уверенным заранее.

```

QTerminal
File Actions Edit View Help
Shell No. 1
[jonni@jbmachine ~]$ python2
Python 2.7.9 (default, Dec 11 2014, 04:42:00)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from test import pystone
>>> pystone.main()
Pystone(1.1) time for 50000 passes = 0.707425
This machine benchmarks at 70678.9 pstones/second
>>>
[jonni@jbmachine ~]$ python
Python 3.4.2 (default, Oct 8 2014, 13:44:52)
[GCC 4.9.1 20140903 (prerelease)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from test import pystone
>>> pystone.main()
Pystone(1.2) time for 50000 passes = 0.952733
This machine benchmarks at 52480.6 pstones/second
>>>

```

► **Тест PyStones скорее всего будет медленнее в Python 3, но это относится не ко всем кодам вообще. Не отказывайтесь от Py3k не глядя!**

Тип `unicode` должен использоваться для взаимодействия с текстом — поиска длины текста, разрезания или переворота строки. Например, код греческой буквы «пи» в нижнем регистре в Unicode равен `U+03c0` в шестнадцатеричном формате. Поэтому задать строку Unicode в консоли Python (если терминал поддерживает вывод в Unicode и использует подходящий шрифт) можно следующим образом:

```

>>> pi = u'U03c0'
>>> print(pi)
π
>>> type(pi)
<набираете в 'unicode'>
>>> len(pi)
1


```

Однако если попробовать сделать это в терминале, не поддерживающем Unicode, ничего не выйдет. Имитировать такой сценарий можно, запустив Python с переменной окружения:

```
$ LC_ALL=C python
```

Если вы теперь попытаетесь напечатать символ «пи» в нижнем регистре, то получите ошибку `UnicodeEncodeError`. По существу, Python пытается принудительно преобразовать этот символ в ASCII (единственный тип, поддерживаемый примитивной локалью C), и у него это не получается. Python 2 также пытается

Скорая помощь



Arch Linux — один из редких дистрибутивов, использующих Python 3 по умолчанию, но он прекрасно уживается и с его предшественником (доступен в пакете `python2`).

» **Не хотите пропустить номер?** Подпишитесь на www.linuxformat.ru/subscribe/

это сделать (независимо от текущих настроек локали) при печати в файл или канал, поэтому не применяйте для таких операций **unicode**: пользуйтесь вместо него **str**.

Тип **str** в Python 2 по сути представляет собой лишь список байтов, соответствующий тому, как строка кодируется на компьютере. Этот тип применяется тогда, когда вы хотите записать строки на диск или отправить их по сети или в канал. Строки типа **unicode** в таких ситуациях Python 2 попытается конвертировать в ASCII (свою кодировку по умолчанию), и это может обернуться слезами. Поэтому, чтобы получить символ «пи», можно воспользоваться его байтовым представлением в UTF-8 напрямую. Существуют правила для преобразования кодов Unicode в байты UTF-8 (или UTF-16), но достаточно просто принять, что в UTF-8 символ «пи» кодируется двумя байтами CF 80. Чтобы Python понял, что это байты, их можно экранировать обозначением `\x`:

```
>>> strpi = '\xCF\x80'
>>> type(strpi)
<наберите 'str'>
>>> len(strpi)
2
```

Так что символ π теперь состоит из двух букв. Учтите, что если в Python 2 вы имеете дело с символами Unicode, то в коде должны быть все обертки и проверки для учета локализации того компьютера, где этот код может запускаться. Также нужно самим

преобразовывать строки между типами **str** и **unicode** и менять кодировку с помощью модуля *codecs*. А если в коде есть строки Unicode, добавьте в начале кода соответствующую декларацию:

```
# -*- coding: utf-8 -*-
```

Основной движущей силой создания новой версии Python была необходимость переписать с нуля работу со строками и их представление, для упрощения этого процесса. Некоторые утверждают, что эта идея провалилась (см. например, тираду Армина Ронахера [Armin Ronacher] в его блоге <http://bit.ly/UnicodeInPython3>), но на самом деле всё зависит от ваших целей. Python 3 полностью избавился от старого типа **unicode**, поскольку в Python 3 теперь в Unicode всё. В типе **str** теперь хранятся коды Unicode, по умолчанию используется кодировка UTF-8 (и не нужен указанный выше декоратор `-*- coding`), а в новом типе **bytes** хранятся байтовые массивы, как в старом типе **str**. Для ввода/вывода данных из файлов новый тип **str** понадобится конвертировать в байты, но это легко сделать с помощью метода **str.encode()**. Для чтения текстовых файлов Unicode откройте их в двоичном режиме `'rb'` и преобразуйте байты в строку с помощью обратного метода **bytes.decode()** (подробности см. на рис. стр. 88 внизу).

Но в Python 3 есть масса и других изменений, кроме этой заморочки с Unicode: одни изменения — это просто удаление старого кода совместимости (Python 3 в отличие от 2.7 не обязан быть совместимым с 2.0), другие предоставляют новые возможности, третьи — заставляют программистов писать код по-другому.

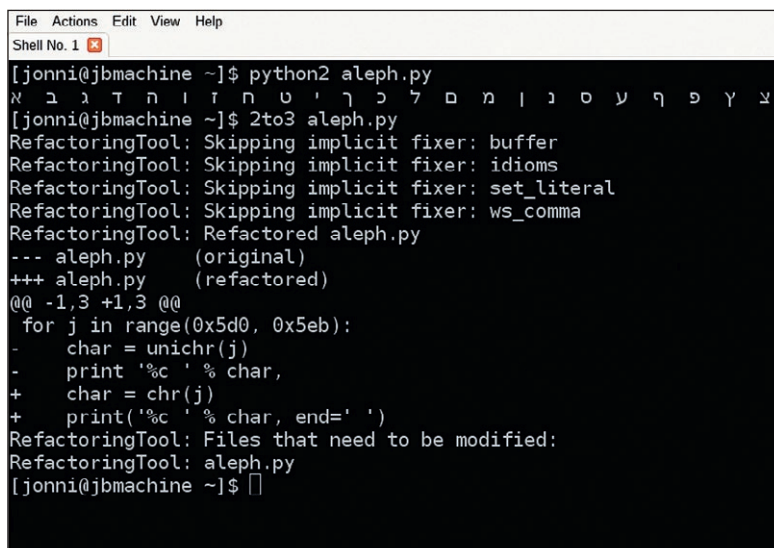
И это еще не все. Например, функциям теперь можно передавать аргументы только по ключевым словам, даже если они используют синтаксис `*args` для списков аргументов переменной длины; а для перехвата исключений с помощью переменных надо использовать ключевое слово **as**. Также для сравнения на равенство можно забыть уродливый оператор `<>` — вместо него теперь более стильный `!=`, который также доступен в Python 2.

Автоматическая конверсия с 2to3

Для достаточно мелких проектов и тех, где не будут встречаться символы из иностранных языков, код на Python 2 можно попробовать автоматически преобразовать в Python 3 с помощью полезной утилиты *2to3*. Она применяет к вашему коду Python 2 заданный набор исправлений, которые позволяют получить настоящий код на Python 3. Для ее использования достаточно запустить

```
$ 2to3 mypy2code.py
```

Эта команда выведет все отличия кода по сравнению с исходным файлом. Также можно воспользоваться параметром `-w` для перезаписи файла, и не волнуйтесь — сначала создается резервная копия. Некоторые исправления применяются только если указать их явно, с помощью параметра `-f`. Пример — исправление



```
File Actions Edit View Help
Shell No. 1
[jonni@jbmachine ~]$ python2 aleph.py
א ב ג ד ה ו ז ח ט י ק ל מ נ ס ע פ צ ף ץ
[jonni@jbmachine ~]$ 2to3 aleph.py
RefactoringTool: Skipping implicit fixer: buffer
RefactoringTool: Skipping implicit fixer: idioms
RefactoringTool: Skipping implicit fixer: set_literal
RefactoringTool: Skipping implicit fixer: ws_comma
RefactoringTool: Refactored aleph.py
--- aleph.py (original)
+++ aleph.py (refactored)
@@ -1,3 +1,3 @@
 for j in range(0x5d0, 0x5eb):
- char = unichr(j)
+ print('%c ' % char,
+ char = chr(j)
+ print('%c ' % char, end=' ')
RefactoringTool: Files that need to be modified:
RefactoringTool: aleph.py
[jonni@jbmachine ~]$
```

➤ **Алеф, бет, гимель...** Утилита *2to3* показывает, как преобразовать код Python 2 для совместимости с Python 3. Шалом алейкем.

Причуды деления

Одно из изменений в Python 3, которое может озадачить — поведение оператора деления `/`. Например, вот фрагмент выполнения кода Python 2:

```
>>> 3/2
1
>>> 3./2
1.5
>>> 3.//2
1.0
```

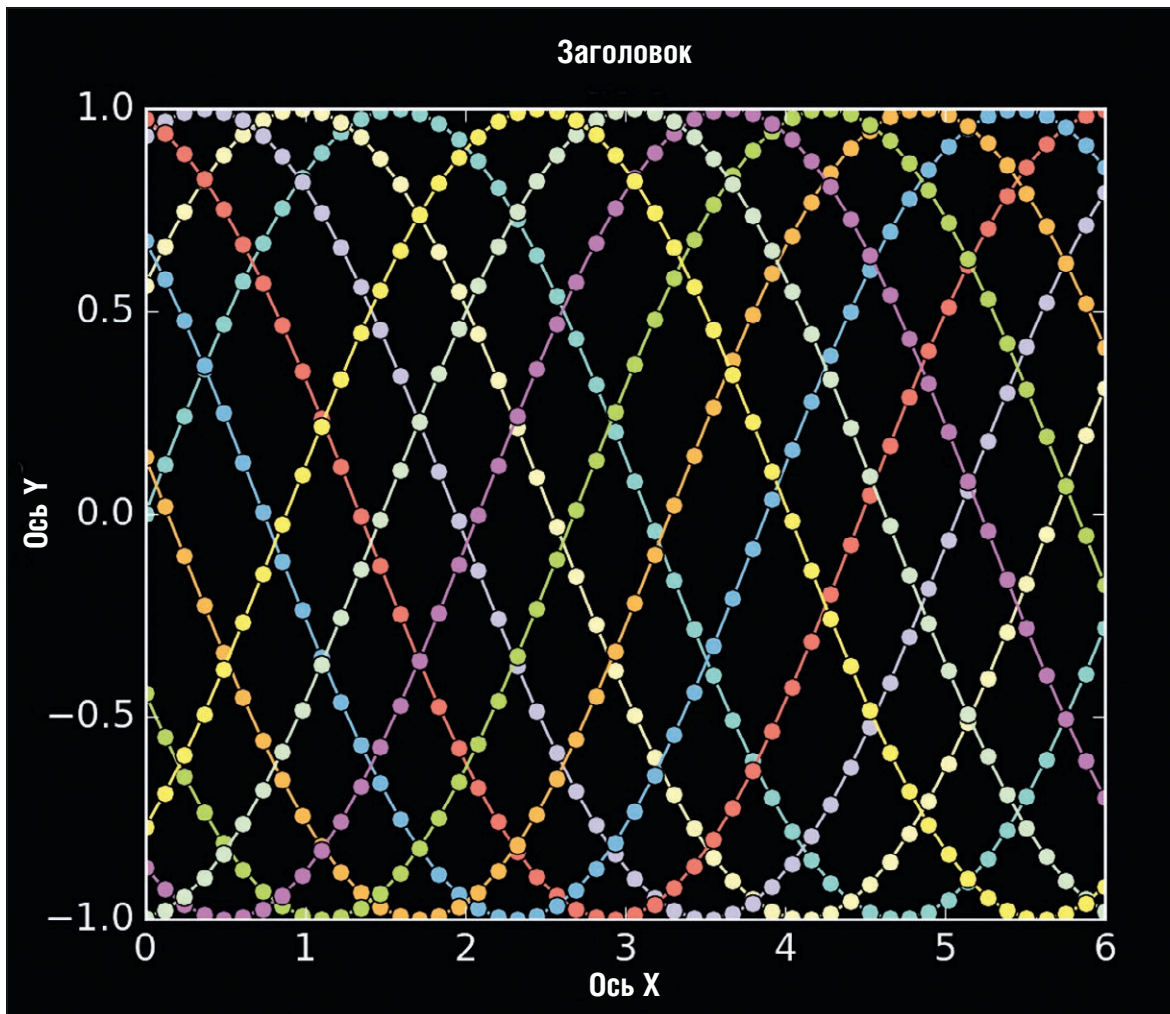
Он показывает, что первая команда выполняет операцию с двумя целыми числами и возвращает

целое число, то есть оператор выполняет целочисленное деление. Во втором примере числитель дробы — число с плавающей запятой, поэтому выполняется вещественное деление, и результат в точности представляет собой половину тройки. В третьей строке используется оператор явного деления с округлением `//`, который возвращает округленный результат в виде вещественного или целого числа, в зависимости от аргументов. Последний оператор был внедрен еще в 2.7, поэтому его поведение в Python 3 не изменилось, но измени-

лось поведение классического оператора деления: если и числитель, и знаменатель — целые, то целое число возвращается, когда числитель делится на знаменатель; иначе возвращается вещественное число с соответствующей десятичной частью. Если как минимум один из аргументов — вещественное (или комплексное) число, поведение не изменяется.

Это означает, что оператор `/` теперь максимально близок к математическому делению, и проблемы с неожиданными целыми результатами больше нас не беспокоят.

➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.



► Популярный модуль *matplotlib* был совместим с Python 3 начиная с версии 1.2, и удовлетворит все ваши требования по построению графиков.

`buffer`, которое заменит все типы `buffer` на `memoryviews`. Эти два типа не полностью совместимы, поэтому для успешной миграции может потребоваться немного поправить код. С помощью *2to3* вы сэкономите массу усилий, так как вручную искать и заменять операторы `print` — не самая приятная задача. В примере на рис. стр. 90 слева показаны изменения в простой программе из трех строк. Функция `unichr()` превратилась в `chr()`, так как Unicode сейчас подразумевается, а строка `print` была переработана, даже если в ней использовались `%` для форматирования выводимых значений.

Говорите ли вы на Python 3?

Другой вариант — написать «двуязычный» код, совместимый как с Python 2, так и с Python 3. Хотя вашей целью должна быть только одна конкретная версия Python, эта промежуточная версия очень удобна при портировании и тестировании кода: вам придется переработать некоторые вещи в Python 2 и при этом наслаждаться новыми возможностями и изучать новый синтаксис Python 3. Многие возможности и элементы синтаксиса Python 3 уже были реализованы в 2.7, и еще больше возможностей доступны с помощью модуля `__future__`. Например, новомодный синтаксис `print` можно получить следующей командой:

```
>>> from __future__ import print_function
```

Этот импорт используется с `from`, а `__future__` ведет себя не как стандартный модуль, а как директива компилятора, которая в данном случае предоставляет синтаксис современного оператора `print`. Аналогично можно импортировать `division` для получения деления в новом стиле или `unicode_literals`, чтобы строки по умолчанию стали формата Unicode. Есть еще один модуль

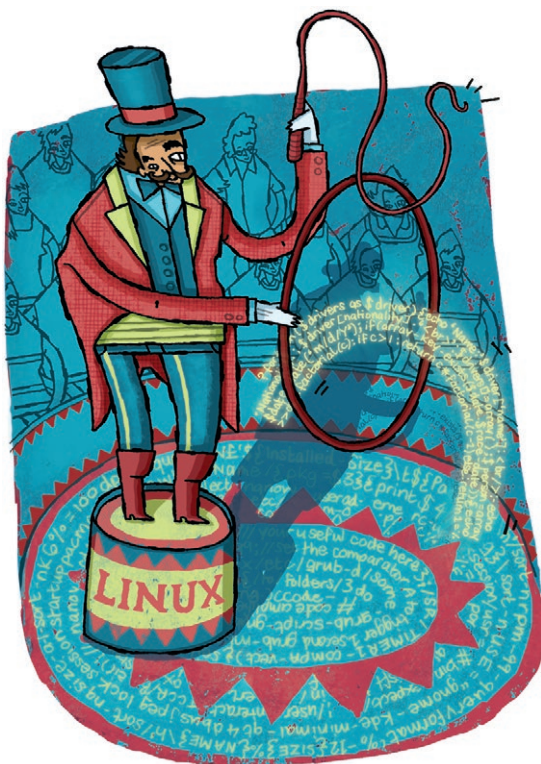
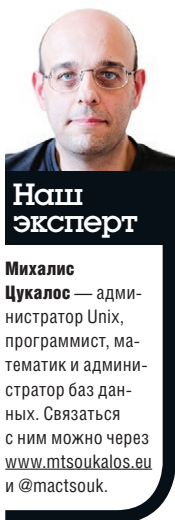
со сбивающим с толку названием `future`, который не входит в стандартную библиотеку, но может упростить проблемы, связанные с переходом.

После выпуска Python 3.0 он, по общим оценкам, был примерно на 10% медленнее своего предшественника. Это было неудивительно, так как скорость не являлась приоритетом новой версии, и многие оптимизации удалили, чтобы очистить исходный код. Теперь, когда у нас есть 3.4 (окончательный релиз 3.5 запланирован на сентябрь 2015 г.), было бы здорово, чтобы производительность улучшилась. К сожалению, этого не произошло, в чем вы можете убедиться сами с помощью теста PyStone. Мы проверили это в офисе [см. стр. 89] на старом компьютере, который дважды возвращался с того света и поэтому обладает сверхъестественными способностями, гораздо большими, чем можно ожидать от процессора с 2,3 ГГц. Но пусть это не вводит вас в заблуждение: PyStone тестирует различные внутренние компоненты Python, которые могут интенсивно использоваться в вашем коде, а могут и не использоваться. Для получения более точной картины важно протестировать ваш код в обеих версиях. Для ускорения кода, где часто используются конструкции C (циклы, математические операции, обращение к массивам) всегда можно воспользоваться Cython [см. «Академия кодинга» LXF193, стр. 88] или модулем `bottleneck`.

Гвидо ван Россум [Guido van Rossum], автор Python, говорит, что Python 2.7 будет поддерживаться до 2020 года, но это не повод отказываться от изучения новой версии сейчас. Python 3 используется все шире, так что вы будете не одиноки. Половина ваших трудностей — в том, чтобы приучить пальцы добавлять круглые скобки. **LXF**

Julia: Динамическое программирование

Михалис Цукалос выстраивает необходимый фундамент, чтобы начать программировать на быстром и динамичном языке Julia.



Julia (<http://julia-lang.org>) — быстрый динамический язык для технических вычислений, спроектированный и разработанный Джеффом Безансоном [Jeff Bezanson], Стефаном Карпински [Stefan Karpinski], Виралем Шахом [Viral Shah] и Аланом Эдельманом [Alan Edelman]. Этот язык также является функциональным и удовлетворяет потребностям высокопроизводительных и научных вычислений, и в то же время может использоваться для общих задач программирования. Для запуска программ в Julia используется JIT-компилятор [JIT — Just In Time, *англ.* точно вовремя] на основе низкоуровневой виртуальной машины (LLVM). Компилятор в сочетании со структурой языка позволяет этому языку приближаться и зачастую достигать производительности C. Функции C также можно вызывать напрямую из Julia без необходимости использовать обертки, специальное API и другие малоудобные штуки.

На структуру и реализацию Julia влияют три основных принципа: во-первых, он должен быть быстрым; во-вторых, динамическим; и в третьих, точным. Ядро Julia написано на C, но остальная часть языка написана на самом Julia, что позволяет вам видеть его работу изнутри и легко вносить изменения.

Даже в «нетипизированных» языках программирования всегда есть некая типизация, используемая компилятором. Julia

```

SRE — mtsouk@mtsouk-VirtualBox: ~ — ssh — 90x45
A fresh approach to technical computing
Documentation: http://docs.julialang.org
Type "help()" to list help topics

Version 0.2.1 (2014-02-11 06:30 UTC)
x86_64-linux-gnu

julia> println('Hello Linux Format!');
ERROR: syntax: invalid character literal

julia> println("Hello Linux Format!");
Hello Linux Format!

julia> typeof("Hello")
ASCIIString (constructor with 1 method)

julia> typeof(4)
Int64

julia> typeof(4.1)
Float64

julia> typeof(typeof)

```

Оболочка Julia — место осуществления большинства ваших экспериментов. Как видите, при делении целых чисел получается результат с плавающей запятой.

думает, что если компилятор использует типы, то почему не разрешить этого программисту, если он захочет? В Julia указание типа переменной оставляется на усмотрение программиста. Однако есть ситуации, когда типы указывать необходимо, поэтому система типизации в Julia широкая и сложная.

Как легко предположить, Julia пытается стать заменой R, Matlab, Octave и модулю NumPy Python в арифметических вычислениях в расчете на лучшую производительность. Реакция других пользователей на эту конкуренцию пока хорошая, и мы покажем, почему.

Установка Julia

В Ubuntu Julia устанавливается командой `sudo apt-get install julia`. После установки ее можно проверить командой

```

$ julia -v
julia version 0.2.1
A версия "Hello World!" в Julia выглядит так:
println("Hello World!")

```

Программу можно сохранить и запустить в оболочке Linux следующим образом:

```

$ julia hw.jl
Hello World!

```

Оболочка Julia — лучшее место для того, чтобы исследовать язык и попробовать понять его возможности (на рис. вверху показано несколько примеров команд в оболочке Julia). Для выхода из оболочки нажмите `Control+d` или наберите `quit()`.

Код Julia можно запускать из оболочки и не сохраняя его. Это очень удобно для маленьких программ:

```

$ julia -e 'for x in ARGS; println(x); end' Hello World

```

```
Hello
World
```

Файл `~/juliarc.jl`, используемый Julia, похож на `.bashrc`. Он выполняет код Julia при каждом запуске оболочки Julia.

```
$ cat ~/juliarc.jl
println("Используем файл juliarc.jl!")
```

Julia — язык программирования, чувствительный к регистру. Имена переменных должны начинаться с буквы (A–Z или a–z), нижнего подчеркивания или других символов Unicode, начиная с 00A0. Поэтому `88a` — некорректное имя переменной, а `_99` — корректное.

В Julia множество типов целочисленных переменных — как знаковых (`Int8`, `Int16`, `Int32`, `Int64` и `Int128`), так и беззнаковых (`UInt8`, `UInt16`, `UInt32`, `UInt64` и `UInt128`), а также булевы переменные (`Bool`) и символы (`Char`). Также поддерживается три типа переменных с плавающей точкой: `Float16`, `Float32` и `Float64`. Число в конце типа означает количество битов, необходимых для данного конкретного типа. В Julia также есть типы `Int` и `UInt`, которые представляют собой алиасы для «родных» целочисленных знаковых и беззнаковых типов системы, в которой запущена Julia. Так, в 64-битной системе `Int` — это алиас для `Int64`.

Знакомимся с ней поближе

Julia также поддерживает дроби, которые записываются с помощью `//`. Так, `4//5` — это дробь, и она не преобразуется в `0.8`. Если у числителя и знаменателя есть общие множители, Julia автоматически сократит дробь, если нет — оставит ее как есть. Дроби могут участвовать в вычислениях с помощью стандартных операторов. Julia также поддерживает комплексные числа. Глобальная константа `im` — это комплексное число `i`. Поэтому `4 - 2im` — комплексное число. Комплексные числа могут участвовать в вычислениях с помощью стандартных операторов.

Теперь давайте рассмотрим какой-нибудь код на Julia. Следующий код вычисляет числа Фибоначчи:

```
$ cat fibonacci.jl
fib(n) = n < 2 ? n : fib(n - 1) + fib(n - 2)
```

Функция `fib()` принимает только один входной аргумент. Как видите, объявлять его тип необязательно. Если захотите, сделать это можно вот так:

```
fib(::Int8) = n < 2 ? n : fib(n - 1) + fib(n - 2)
```

Если вы сохранили свой код в отдельном файле, можете воспользоваться следующей командой:

```
julia> include("fibonacci.jl")
fib (generic function with 1 method [общая функция с 1-м методом])
julia> fib(10)
55
> fib("test")
```

Модуль Profile

Инструменты модуля Profile способствуют отладке кода. При запуске кода модуль производит измерения, помогая понять, сколько времени идет на выполнение каждой строки.

Вы запускаете профилируемую функцию один раз для компиляции и затем снова, вставив ключевое слово `@profile`. И, наконец, командуete `Profile.print()`, чтобы получить вывод. Julia в быстром темпе.

```
ERROR: `isless` has no method matching [нет подходящего метода]
isless(::ASCIIString, ::Int64)
in fib at /Users/mtsouk/docs/article/working/Julia.LXF/code/fibonacci.jl:2
julia> fib(' ')
2178309
julia> fib(32)
2178309
julia> @elapsed fib(40)
0.883197884
```

Хотя вам не удастся определить число Фибоначчи для строки "test", для одного символа вы сможете это сделать — Julia автоматически употребит ASCII-код символа, указанного в качестве аргумента функции `fib()`! Макрос `@elapsed` вычисляет выражение и возвращает количество секунд, потраченное на его вычисление, в виде числа с плавающей точкой, отбрасывая результирующее значение. Это очень удобно для оценки производительности кода.

Два других полезных макроса — `@linux` и `@unix`. Они помогут определить операционную систему, где запущена программа, и выполнить соответствующие действия:

```
julia> @linux? println("Linux!") : println("Not Linux!")
Not Linux!
julia> @unix? println("UNIX!") : println("Not UNIX!")
UNIX!
Not Linux!
```

При всей своей простоте, Julia также подходит опытным программистам. С помощью функции `code_native()` можно увидеть код ассемблера, который компилятор JIT генерирует для функции `println()` [см. рис. на стр. 94 внизу]. Если вы разбираетесь в Ассемблере, то ясно поймете, что код на Julia может выполняться почти так же быстро, как код на C, несмотря на то, что это язык с преимущественно динамической типизацией. Такой оптимизированный код нельзя с легкостью сгенерировать для Python или Ruby, поскольку компилятору JIT не известен тип передаваемых аргументов.

Скорая помощь

Julia поддерживает сокеты TCP. Просто выполнив вызов `listen(1234)`, можно создать сокет, который слушает порт 1234 на localhost. Для подключения его к Julia наберите `connect(1234)`. Если вы использовали присваивание `client = connect(1234)`, то затем сможете закрыть сокет, набрав `close(client)`.

Зачем создавали Julia?

Создатели Julia говорят: «Мы хотели создать открытый язык со свободной лицензией. Мы хотели добиться скорости C с динамичностью Ruby. Мы хотели создать универсальный язык с настоящими макросами, как в Lisp, но и с очевидными и знакомыми математическими обозначениями, как в Matlab. Мы хотели получить нечто подходящее для общего программирования, как Python, легко применимое для статистических расчетов,

как R, с естественной обработкой строк, как в Perl, такое же мощное для линейной алгебры, как Matlab и так же хорошо склеивающее программы вместе, как оболочка. Нечто очень простое для изучения, но и удовлетворяющее большинство серьезных программистов. И чтобы язык был и интерактивным, и компилируемым. (Мы сказали, что он должен быть таким же быстрым, как C?)» Короче, мало они не хотели. Одна из целей Julia —также

улучшение, ускорение и повышение эффективности числового кода по сравнению с другими языками программирования. Библиотека, в основном написанная на самом Julia, включает зрелые библиотеки C и Fortran для линейной алгебры, генерации случайных чисел, обработки сигналов и строк. Вдобавок к этому сообщество Julia предоставляет большое количество внешних пакетов, доступных через встроенный менеджер пакетов.

» Не хотите пропустить номер? Подпишитесь на www.linuxformat.ru/subscribe/

Для вызова функции на C используется функция `ccall()`. Следующий код вызывает системный вызов `getpid()` для получения идентификатора вызывающего процесса:

```
julia> pid = ccall(:getpid, "libc", Int32, ())
44763
julia> typeof(pid)
Int32
```

Большинство операторов в Julia — это просто функции, поэтому $1+2+3$ эквивалентно `+(1,2,3)`. Можно выполнять и такие выражения, как `+(-1,2,3)`. В Julia также поддерживаются анонимные функции, и их основное назначение — передавать их в качестве аргументов другим функциям. Функции в Julia могут возвращать несколько значений и другие функции.

Вот программа, которая считывает целое число, введенное пользователем, и возвращает его множители:

```
function factors(n)
    f = [one(n)]
    for (p,e) in factor(n)
        f = reduce(vcat, f, [f*p^j for j in 1:e])
    end
    return length(f) == 1 ? [one(n), n] : sort!(f)
end
```

Результат функции `factors()` — массив множителей. Для получения данных от пользователя можно воспользоваться функцией `input()`:

```
function input(prompt::String="")
    print(prompt)
   .chomp(readline())
end
```

Для автоматической обработки ввода пользователя можно употребить функцию `factors()` совместно с `input()`. Это делается таким образом:

```
julia> factors(int(input("Give me a number: ")))
```

Функция `readline()` позволяет получить данные пользователя и вывести их на экран функциями `print()` или `println()`. Различие между этими функциями в том, что `println()` автоматически выводит в конце символ новой строки.

В Julia есть возможность обработки ошибок. Следующий фрагмент покажет, как обрабатывать ошибку деления на ноль при делении целых чисел:

```
julia> try
    div(100, 0)
catch x
    println(typeof(x))
end
```

DivideError

Массивы в Julia являются первостепенными объектами, с богатой функциональностью. Командой `myVec = rand{Int32, 100}` легко создать новый массив из 100 случайных целых чисел с именем `myVec`. Для обращения к первому элементу `myVec` используется конструкция `myVec[1]`; стало быть, индекс первого элемента — 1, а не 0. Тогда последним элементом будет `myVec[100]`, а не `myVec[99]`. Для обращения к последнему элементу списка или массива также можно указать в качестве индекса символ конца. Может показаться, что массивы в Julia ничем не отличаются от массивов в других языках программирования. Подлинные преимущества Julia раскрываются в многомерных массивах:

```
julia> anArrayA = [-1 -2 -3; 1 2 3; 10 20 30] # Команда 1
julia> anArrayB = [1 2 3; -1 -2 -3; -10 -20 -30] # Команда 2
julia> arrayAPlusB = anArrayA + anArrayB # Команда 3
julia> productAB = anArrayA * anArrayB # Команда 4
```

Первая команда создает двумерный массив `anArrayA`. Вторая создает второй массив под названием `anArrayB`. Третья команда складывает два массива вместе и сохраняет результат в переменной `arrayAPlusB`. Аналогично, четвертая команда находит произведение двух массивов и сохраняет его в новой переменной. То есть операции с матрицами записываются как обычные арифметические операции.

Команда `anArrayA + 2` добавляет 2 к каждому элементу массива! Команда `sum(anArrayA, 1)` вычисляет суммы массива по столбцам: $(-1 + 1 + 10)$, $(-2 + 2 + 20)$ и $(-3 + 3 + 30)$.

Julia также поддерживает разреженные матрицы. Разреженная матрица — это матрица, большинство элементов которой равны нулю. Для экономии места у таких матриц хранятся только значения и расположения ненулевых элементов.

Строки в Julia

Строки записываются в двойных кавычках и могут содержать любые символы Unicode. Одиночные символы записываются в одинарных кавычках. В Julia есть обычные функции для преобразования строк в верхний или нижний регистр — `uppercase()` и `lowercase()` соответственно. Функция `length()` определяет длину строки. Для объединения строк используется оператор `*`, а не `+`. Также строки можно объединить с помощью функции `string()`.

Чтобы найти внутри строки символ или другую строку, можно воспользоваться функцией `search()`. Функция `replace()` имеет три аргумента и заменяет одну строку другой. `Search()` также поддерживает регулярные выражения. Еще три удобных функции — `repeat()`, `matchall()` и `eachmatch()`, и их использование показано в следующих примерах:

```
julia> thisx3 = repeat("This ", 3)
"This This This "
julia> "This" ^3 # То же, что repeat("This",3)
"ThisThisThis"
```

Внешние программы

Julia также позволяет запускать внешние программы и получать результаты их запуска. Для запуска команды `ls` наберите `run("ls")`. Чтобы присвоить ее вывод новой переменной, наберите `lsOutput = readall("ls")`. Если вы хотите запустить в оболочке переменную `command`, поставьте перед ней знак `$`, например: `run("$command")`. Учтите, что это небезопасный способ написания программ!

➤ **Посмотреть код ассемблера для вашего кода на Julia просто. Например, вот что получилось из функции `println()`.**

```
SRE - mtsouk@mail: ~ - julia - 90x45
julia> code_native(println, (ASCIIString,))
.section
__TEXT,__text,regular,pure_instructions
Filename: string.jl
Source line: 8
push    RBP
mov     RBP, RSP
push   RBX
sub     RSP, 40
mov     QWORD PTR [RBP - 48], 6
Source line: 8
movabs RBX, 4543793712
mov     RAX, QWORD PTR [RBX]
mov     QWORD PTR [RBP - 40], RAX
lea    RAX, QWORD PTR [RBP - 48]
mov     QWORD PTR [RBX], RAX
xorps  XMM0, XMM0
movups XMMWORD PTR [RBP - 32], XMM0
mov     QWORD PTR [RBP - 16], 0
movabs RAX, 140513420375024
Source line: 8
mov     RAX, QWORD PTR [RAX]
mov     QWORD PTR [RBP - 24], RAX
test   EDX, EDX
je     55
```

➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.


```
julia> r = matchall(r»[\w]{4,}», thisx3)
3-element Array{SubString{UTF8String},1}:
"This"
"This"
"This"
julia> r = eachmatch(r»[\w]{4,}», thisx3)
RegexMatchIterator{Regex{UTF8String},String{UTF8String},false}
julia> i = 1 # сперва надо объявить
1
julia> for(word in r)
println("Match ", i)
i += 1 # i++ не поддерживается
end
Match 1
Match 2
Match 3
```

Функция **matchall()** возвращает массив соответствий регулярному выражению для каждого соответствия, а функция **eachmatch()** возвращает итератор по всем соответствиям.

Реализация пузырьковой сортировки

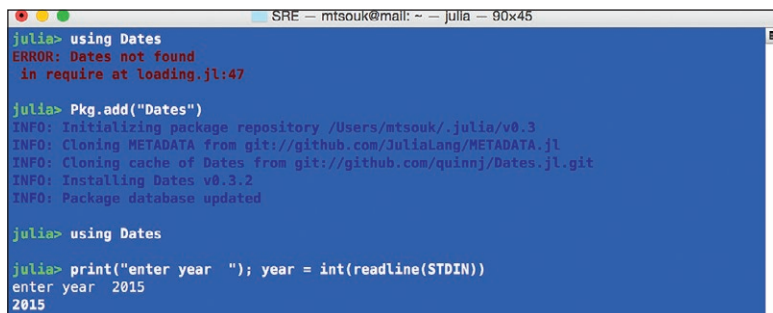
Алгоритм пузырьковой сортировки в Julia, как вы увидите ниже, имеет несколько иной формат по сравнению с циклом **for** в C:

```
function bubbleSort(inputVec)
elements = length(inputVec)-1
for veclIndex in elements:-1:1
for pass in 1:veclIndex
# Если надо, поменяйте местами
if inputVec[pass] > inputVec[pass + 1]
tmp = inputVec[pass]
inputVec[pass] = inputVec[pass + 1]
inputVec[pass + 1] = tmp
end
end
end
return inputVec
end
```

Чтобы получить числа **veclIndex** в обратном порядке, используйте формат **elements:-1:1**. Комментарии в Julia начинаются с решетки.

Теперь вы сможете отсортировать массив **myVec** функцией **bubbleSort()**, выполнив следующую команду:

```
print(bubbleSort(myVec))
```



```
julia> using Dates
ERROR: Dates not found
in require at loading.jl:47

julia> Pkg.add("Dates")
INFO: Initializing package repository /Users/mtsouk/.julia/v0.3
INFO: Cloning METADATA from git://github.com/JuliaLang/METADATA.jl
INFO: Cloning cache of Dates from git://github.com/quinnj/Dates.jl.git
INFO: Installing Dates v0.3.2
INFO: Package database updated

julia> using Dates

julia> print("enter year "); year = int(readline(STDIN))
enter year 2015
2015
```

Для показа информации об установленных пакетах пригодится команда **Pkg.status()**. (На рис. справа сверху показано, как установить и воспользоваться пакетами в Julia.) Для добавления пакета используется команда **Pkg.add()**, для удаления — **Pkg.rm()**. Наконец, команда **Pkg.update()** позволит обновить каждый установленный пакет до последней и наилучшей версии.

В Julia также можно взглянуть на исходный код и полную историю разработки всех установленных пакетов, изменить их код и исправить ошибки в нем. Так, например, исходный код пакета *Dates* можно найти в каталоге `~/julia/v0.3/Dates/`. Возжелав создать собственный пакет, можете начать с команды **Pkg.generate()**.

Julia позволяет работать и с текстовыми, и с двоичными файлами. В этом разделе мы посмотрим, как работать с текстовыми файлами. Представленная программа считывает текстовый файл строка за строкой, преобразует каждую строку в верхний регистр и сохраняет вывод в новый текстовый файл. Код выглядит так:

```
input = ARGS[1]
output = ARGS[2]
OUTPUT = open(output, "w")
open(input, "r") do f
for line in eachline(f)
print(OUTPUT, uppercase(line))
end
end
close(OUTPUT)
```

При чтении файла с помощью **do** файл автоматически закрывается при достижении Julia конца файла. В других случаях файл нужно закрывать командой **close()**.

В Julia гораздо больше возможностей, чем представлено в этой статье, и, как и во всех современных языках программирования, здесь много удобных функций, преимущественно для научных расчетов. **LXF**

» Вот как установить пакет *Date* в Julia для получения даты.



Построение графиков в Julia

TextPlots — невероятно простая библиотека для построения графиков, которая строит графики в терминале с помощью символов азбуки Брайля. Это позволяет построить график любой непрерывной функции с действительными значениями и любого относительно небольшого набора точек данных.

На рисунке (справа) показан вывод следующих команд:

```
julia> Pkg.add("TextPlots")
julia> using TextPlots
julia> plot(cos)
julia> plot([x->cos(x) * sin(x), x->x/5], -6:6)
```

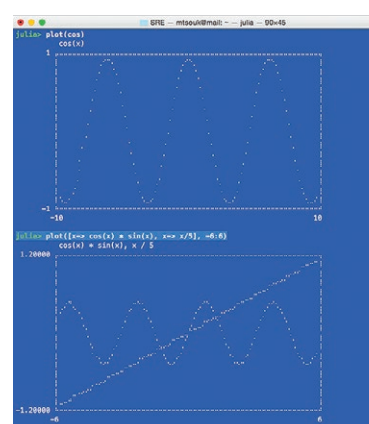
Как вы понимаете, *TextPlots* позволяет одновременно выводить несколько кривых

и определить диапазон значений *x*. В следующем коде с помощью разных вариантов вызова команды **plot()** мы получаем три различных графика:

```
julia> plot(cos, border=false)
julia> plot(cos, border=false, title=false, labels=false)
julia> plot([1, 2, 5, 8], [10, 11, 12, 3])
```

Последняя команда выводит на экран с помощью команды **plot()** пары точек (*x*, *y*).

Более подробную информацию об использовании пакета *TextPlots* в Julia можно найти на <https://github.com/sunetos/TextPlots.jl>. Другие удобные пакеты для построения графиков, которые вы можете попробовать — *Winston*, *PyPlot* и *Gadfly*.



» Пакет *TextPlots* позволяет строить графики в терминале с помощью символов азбуки Брайля, и это очень удобно, когда нет графического интерфейса.

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...

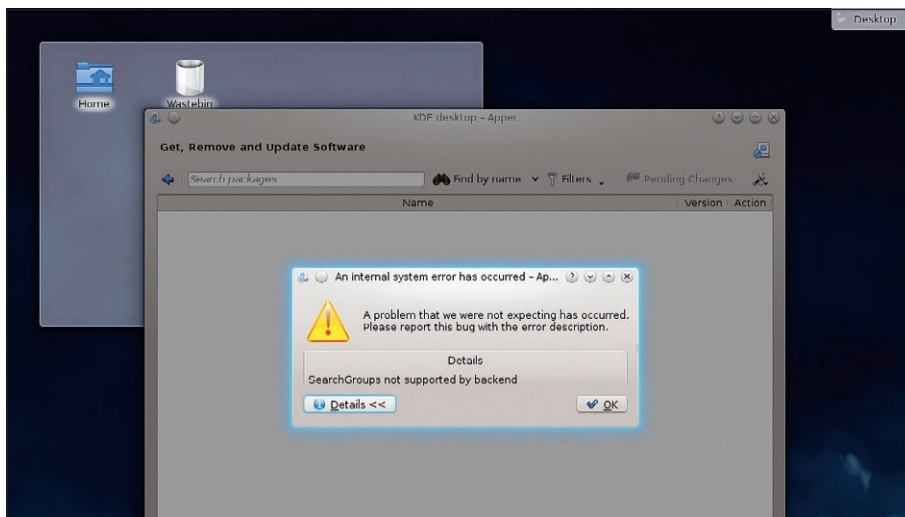
- 1 Менеджер пакетов в Fedora 21
- 2 Горячие клавиши с дурным поведением
- 3 Резервное копирование DVD
- 4 Копирование DVD Linux Format на флэшки
- 5 Загрузку при наличии UEFI
- 6 Дело об исчезнувших принтерах

1 Apper не работает

В Недавно я загрузил и установил дистрибутив из образа Fedora-Live-KDE-i686-21-5.iso. Установка прошла гладко, и я успешно и безо всяких проблем установил все доступные обновления. Меня впечатлило то, как быстро и ровно все это прошло. Однако в системе есть проблема — невозможно установить дополнительные пакеты из стандартного репозитория Fedora. При выборе System Settings [Настройки системы] и затем Software Management [Управление ПО] я попадаю в раздел Lists and Groups [Списки и группы]. При нажатии на любую из иконок в разделе Groups [Группы] система выдает окошко с сообщением “A problem that we were not expecting has occurred. Please report this bug with the error description [Возникла неожиданная проблема. Пожалуйста, сообщите о ней с описанием ошибки]”. Если нажать в этом окне на кнопку Details [Подробности], оказывается, что поиск по группам не поддерживается сервером. Я не могу понять этой ошибки. Правильный ли файл ISO я загрузил? В имени файла есть i686; правильно ли я думаю, что P1 — i586, а P11 — i686? Можно ли как-то включить поиск по группам на сервере?

stuarте9, с форумов

Вы правильно думаете, что i686 — это 32-битный P11 или позже. 64-битные ISO обычно обозначаются как amd64 или x86_64. Это известная проблема: она выражается в том, что Apper, менеджер пакетов KDE для Fedora, пытается вызвать функцию, которая не реализована в серверной части управления пакетами. Здесь одна из тех ситуаций, когда ошибку всегда велят на кого-то другого, в зависимости от того, кого Вы спрашиваете: «ошибка серверной части в том, что функция не реализована» или «ошибка клиентской части в том, что она предположительно не выбрала другую альтернативу». Как бы то ни было, ошибку в конце концов придется исправлять, но до этого момента функции групп в Apper работать не будут. А пока ошибка



» Менеджер пакетов KDE в Fedora 21 «падает» при попытке просмотра групп. В качестве временного решения проблемы воспользуйтесь другим менеджером пакетов, таким как yumex.

не исправлена, Вам нужно воспользоваться альтернативным менеджером пакетов — например, yum в командной строке или другим, графическим менеджером. Если Вы взяли образ с LXF DVD194, то там были клиенты для Gnome и для KDE, и Вы уже можете воспользоваться gnome-software. Так как Вы установили дистрибутив с образа только для KDE, можете установить gnome-software из командной строки, например, так:

```
yum install gnome-software
```

после чего поискать дополнительные пакеты в gnome-software. Поскольку это программа для Gnome, ее установка в KDE притянет за собой огромное количество зависимостей. В качестве альтернативы можно воспользоваться YUM Extender (yumex), менеджером пакетов из LXDE, установка которого требует меньше зависимостей. Вы увидите, что при просмотре групп в gnome-software не возникает таких проблем, как в Apper. Эта проблема относится только к группам. Обновление, поиск и установка пакетов по имени в Apper прекрасно работают.

2 Скрипты и горячие клавиши

В У меня Xubuntu 14.10. Я установил xclip и написал однострочный скрипт для чтения системной даты:

```
date "+ %A %d %B %Y" | xclip -selection clipboard
```

Я также написал однострочный скрипт для вывода даты на экран:

```
xclip -o -selection CLIPBOARD
```

Оба скрипта работают из командной строки. Удобства ради я настроил горячие клавиши для

каждого из этих скриптов. Скрипт, который считывает дату, при запуске с горячих клавиш работает хорошо, а скрипт, который должен выводить дату, кажется, ничего не делает. У меня есть ощущение, что я не замечаю чего-то простого, но не могу понять чего.

elenmar, с форумов

Диагностика таких проблем бывает затруднительной, поскольку при запуске с горячих клавиш Вы не видите никаких сообщений команды, а при запуске в терминале изменяется среда выполнения скрипта. Если Вы открываете сеанс оболочки в терминале, та запускает Ваш профиль, обычно ~/.bash_profile или ~/.bashrc, тогда как при запуске скрипта комбинацией клавиш (или как задания cron) этого не происходит.

Ну и как же тогда увидеть вывод скрипта, если он не связан с окном терминала? Добавьте такие строки в начало скрипта:

```
exec >>-/myscript.log 2>&1
set -x
```

Изменить направление вывода для процесса оболочки можно командой exec; приведенная выше строка перенаправляет стандартный поток вывода и стандартный поток ошибок в файл. Первое перенаправление направляет стандартный вывод в файл, затем 2>&1 направляет стандартный поток ошибок в стандартный поток вывода. Теперь Вы сможете просмотреть вывод программы, даже вызывая ее горячими клавишами. Для просмотра вывода в реальном времени запустите в терминале команду

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`.

```
tail -f ~/myscript.log
```

и затем нажимайте горячие клавиши. Вторая строка `set -x` сообщает оболочке, что следует печатать каждую команду перед запуском; с ее помощью Вы увидите, что именно выполняет скрипт и какие команды формируют все сообщения.

Если есть подозрение, что проблема вызвана различием сред выполнения, можно заставить скрипт загружать профиль перед запуском любых команд. Для этого используется одна из следующих команд:

```
source ~/.bash_profile
source ~/.bashrc
```

Наконец, можно написать один скрипт, который просто выводит дату в необходимом формате, но, подозреваю, у Вас имеется причина сохранять эту дату, чтобы использовать ее в дальнейшем. Учтите, что содержимое буфера обмена может изменить или очистить любая программа, поэтому, возможно, безопаснее использовать временный файл, чем буфер обмена.

3 Резервное копирование DVD

В Я хочу скопировать, сохранить и сделать резервную копию нескольких DVD-дисков из моей коллекции на NAS, чтобы затем посмотреть их с помощью Raspberry Pi и XBMC. В идеальном случае, я хотел бы сохранить видео в форматах MKV или MP4. Похоже, мне понадобится удалить защиту от копирования, сделать копию диска в ISO и преобразовать полученный файл в MKV или MP4. Какие программы вы можете посоветовать для этого?

Noob_Computa_Ninja, с форумов

Прежде всего нужно решить, какую часть каждого DVD Вы хотите скопировать (учтите, что в Вашей стране может быть запрещено копировать DVD или снимать защиту DRM; в Великобритании на данный момент

это разрешается делать для личного пользования). Вам нужен весь DVD с меню и дополнительными роликами или только основной фильм? Если Вам нужен весь DVD и в проигрывателе установлен `libdvdcss`, можно просто скопировать весь DVD в файл ISO следующей командой:

```
cp /dev/dvd name.iso
```

Так Вы получите самую точную копию, но платой за нее станет большой размер файла — каждый образ может иметь размер до 9,4 Гб. Если Вам нужен только файл фильма с каждого DVD, то есть несколько программ, которыми Вы сможете воспользоваться — как для командной строки, так и с графическим интерфейсом. Выбор программы опять же зависит от Ваших потребностей. Если Вам нужна просто копия файла без перекодировки, можете воспользоваться опцией `dump` в `MPlayer`.

```
mplayer -dumpstream -dumpfile movie.mpg dvd://1
```

Эта команда вместо воспроизведения копирует первую дорожку DVD в файл. Единственная обработка, которая происходит — удаляется любое шифрование. Если Вы хотите перекодировать видео в другой формат, хорошая графическая программа для этой задачи — `HandBrake` (инструмент командной строки у нее тоже есть). Она должна быть в репозиториях большинства дистрибутивов, так что можете установить ее обычным образом и запустить. Хотя в `HandBrake`, кажется, нет возможности напрямую читать DVD, но нажав кнопку `Source [Источник]` и выбрав `/dev/dvd` (или `/dev/sr0`, в зависимости от Вашей системы), можно прочитать DVD, находящийся в этом приводе, выбрать фильм для перекодирования и указать параметры кодирования. Это важно



Коротко про...

Поиск файлов

В Linux есть две основные утилиты для поиска файлов: `locate` и `find`. Первая работает с базой данных файлов, которая хранится в системе. Она очень быстрая, но ограничена файлами, существовавшими на момент последнего обновления базы данных. В большинстве дистрибутивов для поддержания базы данных в актуальном состоянии применяется скрипт `cron`. `Locate` умеет искать только по именам файлов. Общий формат ее использования таков:

```
locate somefile
```

Добавьте `-i`, чтобы поиск был нечувствителен к регистру. Ее альтернатива — утилита `find` — ищет файлы непосредственно в файловой системе. Она медленнее и может найти только те файлы, для которых у пользователя есть права на чтение, но предоставляет самые актуальные результаты. Она позволяет указать, в каких каталогах нужно искать, и выполнять поиск

по другим параметрам, кроме имени файла: например, владельцу или дате создания.

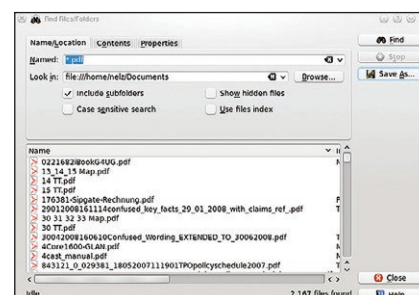
```
find -name '*somefile*'
find /usr -iname '*someotherfile*'
find /usr -maxdepth 2 -iname '*whatfile*'
```

Первая строка начинает поиск в текущем каталоге и опускается во все подкаталоги. Вторая начинает поиск в `/usr` и выполняет его с учетом регистра. Третья делает то же самое, но углубляется не ниже чем на два каталога.

`Find`, при ее огромном количестве параметров, гораздо более гибкое решение, но `locate` прекрасно подходит для быстрого поиска. Для поиска в заданных каталогах вывод `locate` можно пропустить через `grep`:

```
locate -i myfile | grep /home/
```

Обратите внимание, что `locate` производит поиск подстроки, а `find` ищет точное соответствие, поэтому в качестве шаблона используется `*`.



> `Kfind` придает дружелюбный вид мощной команде `find`.

Вам когда-нибудь было интересно, где программа хранит свои конфигурационные файлы? Выполните команду

```
touch /tmp/now
```

затем запустите программу, измените ее настройки и закройте ее. Выполните команду:

```
find ~ -newer /tmp/now
```

чтобы найти все файлы, которые были изменены; это конфигурационные файлы программы и, возможно, какие-то еще.

учитывать, если для воспроизведения используется Raspberry Pi. Следует выбирать кодеки и сжатие, которые будут совместимы с Pi. К счастью, в *HandBrake* есть несколько функций, которые с этим помогут. Для проверки можно перекодировать небольшую часть видео с разными параметрами (а не весь фильм) и сохранить эти настройки в пресет для быстрого использования в будущем.

Можете также начать с пресета Normal и оставить видекодек H.264, но аудиокодек заменить на AAC. Попробуйте перекодировать короткий фрагмент и посмотрите, как он будет воспроизводиться. Затем, если что-то не так, можно изменить настройки. Определившись с настройками, нажмите на иконке Save [Сохранить] под списком Presets [Пресеты].

Если Вы хотите перекодировать несколько дисков и у Вас достаточно места, можно сохранить каждый диск в файл ISO как раньше, а затем настроить перекодирование в *HandBrake* и добавить диски в очередь на копирование. После этого выберите Queue > Start Queue [Очередь > Начать обработку очереди], и все диски будут скопированы.

4 DVD Linux Format в USB

В Не подскажете, почему у меня не получается сделать загрузочную флэшку из DVD LXF192? Я пробовал сделать это на двух разных 64-битных компьютерах. Я загрузился с диска LXF192, загрузил Ubuntu 14.10 Mate и воспользовался создателем диска [disc creator]. Минут через 20–30 он выдал ошибку, и процесс остановился. Я попробовал снова, на 64-битном ноутбуке Compaq HP, очистил флэшку, и через 20–30 минут произошло то же самое. В чем причина? У меня никогда не получается скопировать на флэшки диски, которые идут с журналом. Это я один такой невезучий?

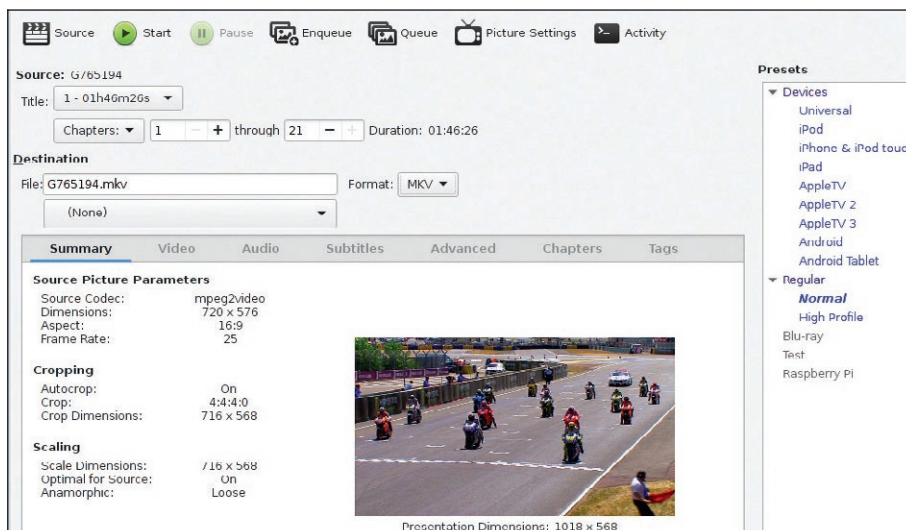
e110, с форумов

Disc creator как тот, которым Вы пользовались, и более общие, такие как *Unetbootin*, делают определенные предположения, которые не всегда оказываются верными. В данном случае он предположил, что Вы загрузились с диска Ubuntu, и ожидал найти его в оптическом приводе. Но на самом деле это совсем другой диск, и Вы загрузились с ISO-образа этого диска.

Этого достаточно для того, чтобы данный подход не сработал, но хорошие новости в том, что решить проблему не просто возможно, но и сделать это гораздо проще, чем возиться со всякими disc creators. Как LXF DVD, так и ISO-образы различных других дистрибутивов известны как гибридные образы. Это означает, что они загружаются как с диска, так и с флэшки. Чтобы скопировать весь DVD на флэшку, выполните следующую команду:

```
sudo dd if=/dev/dvd of=/dev/sdX bs=4k sync
```

Обратите особое внимание на то, чтобы точка монтирования была обязательно указана правильно, поскольку если Вы укажете не то



HandBrake позволяет легко перекодировать видео и задать собственные пресеты.

устройство, все данные на нем будут уничтожены. Чтобы не ошибиться, выполните команду

```
dmesg -w
```

а уж потом вставляйте флэшку. Вы должны увидеть нечто вроде этого:

```
usb-storage 1-2:1.0: USB Mass Storage device detected [Устройство хранения обнаружено]
scsi host7: usb-storage 1-2:1.0
scsi 7:0:0:0: Direct-Access SanDisk Cruzer Edge 1.20
PQ: 0 ANSI: 5
sd 7:0:0:0: [sdb] 31266816 512-byte logical blocks:
(16.0 GB/14.9 GiB)
```

Здесь это /dev/sdb. Если имя устройства указывается без завершающего номера, то данные записываются на все устройство. Так мы получим копию всего DVD, поэтому процесс будет довольно медленным. Другой вариант — скопировать только тот ISO-образ, который Вам нужен. В Вашем примере Вам нужен только образ Ubuntu, и можно выполнить ту же команду, но указать путь до образа ISO, а не до всего диска:

```
sudo dd if=/media/LXFDVD192/ubuntu-14.10-desktop-remix-amd64.iso of=/dev/sdX bs=4k sync
```

Команда **sync** гарантирует, что все данные будут записаны на флэшку до возвращения в командную строку, чтобы Вы не отключили флэшку слишком рано. В любом случае, флэшку не стоит монтировать во время записи на нее.

5 Загадка UEFI

В Я прочел статью Нейла Ботвика о настройке Grub [LXF193, стр. 64], однако она не помогла решить мою проблему. Я купил Lenovo Flex 10, так как мне всегда нравились компьютеры Lenovo. В отличие от предыдущих моделей, на этой заставить Linux работать было непросто. На ней 32-битная прошивка UEFI и 64-битный процессор. Это значительно затрудняет установку, хотя и не делает ее невозможной. Я дошел до загрузки Ubuntu 13.04 с pendrive, который я подготовил в Windows с помощью Rufus. С него я запустил GParted и настроил свой диск так:

```
Disk partition table set as GPT:
```

Partition	Filesystem	Size	Flag	Mountpoint
/dev/sda1	FAT32	511MB	boot	
/dev/sda2	EXT4	10GB	none	/
/dev/sda3	Swap	1GB	none	

Однако с жесткого диска загрузиться я не могу. Он не распознается прошивкой, и я получаю панику ядра. В консоли Grub я использую следующие строки:

```
linux (hd1,gpt2)/boot/vmlinuz-3.8.0-19-generic root
root=/dev/sda/2
initrd (hd1,gpt2)/boot/initrd img-3.8.0-19-generic
```

Смогу ли я избежать всего этого, воспользовавшись загрузчиком EFI? Сейчас у меня не ноутбук, а недорогое пресс-папье!

Питер Вулли [Peter Woolley]

Начало работы с UEFI может быть немного обескураживающим, особенно если у Вас 32-битный UEFI. Есть несколько важных моментов, о которых нужно помнить. Первый — системный раздел EFI недостаточен просто отформатировать в FAT, его нужно сделать таким в таблице разделов, установив тип раздела в EF00. Также, похоже, имеет место некоторая путаница с определением диска. На ноутбуке, насколько я понимаю, диск должен быть один, /dev/sda на языке Linux, как записано в таблице разделов, но в терминах Grub это hd0. Проект Grub 2 принял достаточно обескураживающее решение, начав подсчитывать диски с нуля, а разделы — с единицы, поэтому sda2 — это hd0.gpt2.

Другой важный момент — это история с кюрицей и яйцом при установке Grub в систему UEFI. Чтобы Grub установился в правильном месте, ему нужна информация об UEFI из прошивки EFI, которую он получает из /sys/firmware/efi/efivars. Если этот каталог пуст или не существует, Grub будет установлен как для загрузки из BIOS, но эта информация доступна только в том случае, если Вы уже загрузились с прошивкой EFI. Поэтому перед установкой Grub убедитесь, что файлы на месте.

Другой важный фактор — Ваше сочетание 32-битного UEFI с 64-битными процессором и ОС.

Ядро Linux поддерживает это, но только начиная с версии 3.15. В Ubuntu 13.04 ядро версии 3.8.8, и оно не поддерживает того, что Вы хотите сделать. При этом, возможно, Вы уже успели прочитать продолжение к этой статье [LXF194, стр. 74], в котором рассказывалось о *Gummiboot* для загрузки EFI.

6 Невидимый принтер

В После обновления с Ubuntu 14.04 на 14.10 у меня никак не получается заставить компьютер видеть мой принтер Officejet 4620, который превосходно работал в 14.04. Это довольно странно, так как если я включаю принтер за минуту или две до включения компьютера, то никаких проблем вроде бы нет. Однако если сначала включить компьютер, а затем принтер, то я вообще не могу печатать, и приходится перезагружать компьютер, чтобы решить эту проблему.

Но это еще не все. Когда я в первый раз попробовал подключить к компьютеру свою камеру, он тоже ее «не увидел». У камеры прямое подключение по USB. Затем я попробовал подключить камеру к другому компьютеру, на котором не было ничего, кроме свежеставленной Ubuntu 14.10. И этот компьютер тоже ее не увидел! Затем я подключил камеру к ноутбуку с Ubuntu 12.04 и спокойно скопировал на него фотографии безо всяких проблем!

Брайан Митчелл (Bryan Mitchell)

Похоже, у Вас проблема с горячим подключением устройств. Первое, что нужно сделать — проверить, видит ли ядро появление принтера вообще. Откройте терминал и выполните команду

```
dmesg --follow
```

Затем подключите принтер и наблюдайте за выводом команды. Определяется ли принтер в *udev*? При включении принтера перед загрузкой системы принтер есть в системе на момент запуска *udev* и CUPS. Откройте web-интерфейс CUPS по ссылке <http://localhost:631> — видим ли принтер? Если видим, но отключен, можете включить его в web-интерфейсе или следующей командой:

```
sudo cupsenable имяпринтера
```

Если CUPS вообще не видит принтера, вы можете, включив принтер, попробовать выполнить команду `sudo /etc/init.d/cups restart`, чтобы заставить CUPS определить его. Это смахивает на колку ореха кувалдой, но кувалда в этом случае гораздо меньше, чем при перезагрузке всего компьютера.

Проблема с камерой, скорее всего, другая, и о ней уже есть запрос на Launchpad (<http://bit.ly/LXF195bugs>) — команда `blkid` неправильно определяет некоторые файловые системы. Вероятно, когда Вы будете читать этот ответ, эта ошибка уже будет исправлена в репозиториях, а если нет, загрузите Deb-файл по ссылке <http://packages.ubuntu.com/vivid/libblkid1> для решения проблемы (ссылка на этот файл есть в комментариях к запросу об ошибке). LXF

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://sourceforge.net/hardinfo.berlios>) — она сохранил 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
```



Часто задаваемые вопросы

Мобильный широкополосный доступ

» Что такое мобильный широкополосный доступ, о котором все болтают? Это что-то вроде ADSL для мобильного телефона?

Нечто вроде этого. Здесь используется мобильная сеть 3G, но технология называется не ADSL, а HSPA (High Speed Packet Access — высокоскоростной пакетный доступ), предназначенная для использования в мобильных устройствах. Эта технология используется в смартфонах, которым нужен постоянный (или по крайней мере частое) подключение к Интернету.

» Он дает скорости на уровне широкополосного доступа?

Нет, он не является широкополосным в исконном смысле слова (как, по правде говоря, не являются такими и медленные проводные соединения), но он достаточно быстр для решения важных задач, таких

как чтение почты и просмотр видеороликов на YouTube. Различные фирмы предлагают различные скорости доступа, но все они сильно зависят от уровня сигнала и других параметров.

» Что мне нужно, чтобы воспользоваться им?

Вам нужен широкополосный модем и договор с оператором мобильной связи. Большинство договоров включают приобретение модема, но можно воспользоваться и собственным. Операторы чаще всего используют одни и те же модемы, USB-брелки, в которые вставляются SIM-карты для идентификации клиентов.

» Сколько это стоит?

Как и с большинством таких вещей, цена зависит от поставщика, продолжительности договора и объема

данных. В Великобритании за £15 можно приобрести около 3 ГБ интернет-трафика в месяц.

» Каков срок действия такого договора?

Длительность договора может составлять от 1 до 18 месяцев; также можно воспользоваться сервисами с оплатой за фактическое потребление трафика. Более длительные договоры предусматривают бесплатный модем.

» Хорошо ли это будет работать в Linux?

Довольно хорошо. Так как все компании предоставляют модемы одного и того же производителя, Huawei, а драйверы для него есть в ядре Linux, он должен «просто работать». Нужно настроить модемное соединение с помощью программы вроде KPPP или

Gnome-PPP, а в качестве устройства модема выбрать `/dev/ttyUSB0`. Так как для аутентификации используется SIM-карта, имя пользователя и пароль могут быть любыми (хотя в программе понадобится их указывать).

» Имеет ли значение, какой у меня дистрибутив?

Стандартные настройки PPP должны работать в любом дистрибутиве. У тех дистрибутивов, которые используют Network Manager, есть преимущество, поскольку теперь он поддерживает 3G-модемы. В таком дистрибутиве после того, как модем вставлен в компьютер, появляется окошко, которое запрашивает у вас провайдера и затем задает все настройки. После этого вы сможете включать и отключать свой модем через меню Network Manager.



LXF HotPicks



Александр Толстой

вручную вымывает самые яркие крупинцы золота в потоке открытого кода, а потом сплавляет их в форму HotPicks.

Yarock » FFmpeg » Dropbox Uploader » Glibus-preview » MPS-Youtube » KWave » Linuxbrew » SuperTuxKart » Aualé » Hollywood » Boomaga

Музыкальный плеер

Yarock

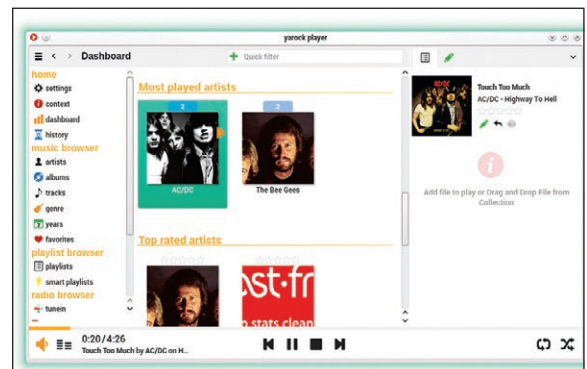
Версия: 1.0.0 Сайт: <https://launchpad.net/yarock>

Хотя музыкальных плееров в Linux вполне хватает, продолжают появляться новые. *Yarock* — еще один новичок, совсем недавно доросший до версии 1.0.0 после пяти лет релизов серии 0.x. Ярчайшая особенность этого плеера, как сказано на его официальном сайте — это «простой, но милый браузер музыкальной коллекции на основе обложек альбомов».

Кроме того, *Yarock* является приложением на Qt4 со стильным видом, который для релиза 1.0.0 был переделан: появились новые значки и кое-что изменилось в структуре. Среди этих изменений — размещение кнопок управления

воспроизведением внизу, а панель категорий в левой части окна плеера теперь содержит больше пунктов.

Интерфейс плеера разделен на три части. В левой панели размещены под-категории (браузеры Home, Music, Playlist и Radio, а также локальные папки), центральная часть показывает, где вы находитесь, а правая панель используется для исследования и, при необходимости,



► Умный и быстрый музыкальный плеер, написанный на чистом C++ и связанный со средами мультимедиа Qt4 и Phonop.

«В Yarock имеется масса вариантов сортировки коллекции.»

изменения метаданных трека. Вероятно, опытных пользователей все эти подробности пока что не потрясли, но прелести *Yarock* проявляются, когда вы начинаете прослушивать в нем свою музыкальную коллекцию. *Yarock* предлагает быструю индексацию файлов в базе данных SQLite3 и позволяет вручную скачать недостающие обложки из Интернета. Перетаскивать песню в список воспроизведения не надо (хотя и можно): *Yarock* воспроизводит ваши треки прямо из подборки. В *Yarock* имеется масса вариантов сортировки коллекции: по альбомам, песням, жанрам, году и даже по папкам. Кроме того, когда вы прослушаете несколько треков, Yarock может сгенерировать умный плей-лист на основе предыстории воспроизведения.

Станет ли *Yarock* вашим постоянным плеером? Дело ваше. Приложение весьма богатое функционально, без зависимостей KDE или Gnome, а для поклонников *Bash* предлагает интерфейс командной строки. Оно вам подойдет, если вам нужен плеер с простой очередью воспроизведения, закладками, редактором тэгов, нормализатором громкости и другими основными функциями.

Кстати, *Yarock* не зависит от рабочего стола — список поддерживаемых им форматов определен движком Phonop. В большинстве случаев он переведет вас на движок *GStreamer*, так что вы уж позаботьтесь о кодеках.

Исследуем интерфейс Yarock

Навигация

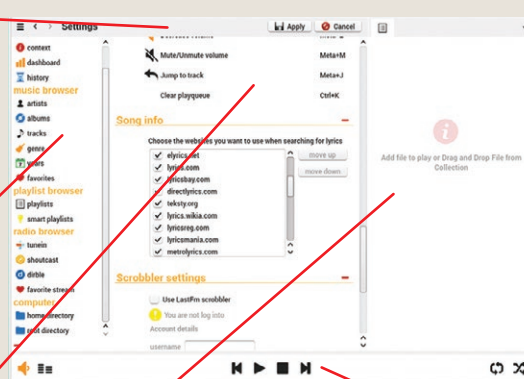
Как и в web-браузере, верхняя панель *Yarock* показывает, где вы находитесь, и размещает дополнительные кнопки, позволяющие принимать решения.

Список функций

Левая панель всегда показывает разные источники музыки и имеющиеся режимы просмотра.

Главный вид

Обычно именно здесь отображается текущий выбор в виде обложки. Этот вид Settings демонстрирует отличные возможности индивидуальной настройки для текстов песен, скроллинга и т. д.



Закрепленное за треком место

Когда что-то воспроизводится, здесь вы можете увидеть подробности. Кроме того, эта панель предназначена для организации пунктов в списке воспроизведения.

Управление воспроизведением

Здесь — кнопки воспроизведения, паузы, остановки и промотки назад и вперед. Дополнительные кнопки сбоку предназначены для управления уровнем громкости, эквалайзером и режимом перестановки треков.

Набор мультимедиа-библиотек и программ

FFmpeg

Версия: 2.5.2 Сайт: <http://ffmpeg.org>

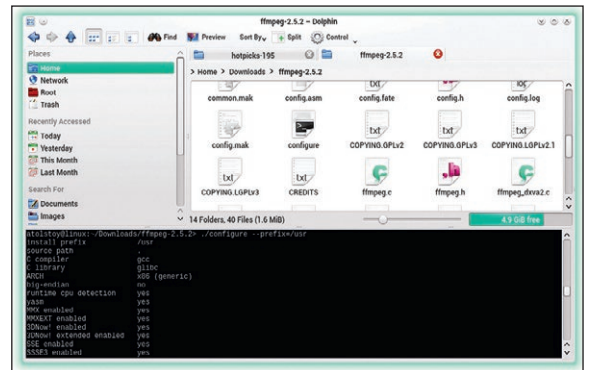
Проект *FFmpeg* весьма зрелый, и признан в тех кругах свободного ПО, где занимаются созданием библиотек и программ для работы с мультимедиа-данными. *FFmpeg* является движком многих популярных мультимедийных плееров: *VLC*, *MPlayer*, *Handbrake Converter* и многих других. *FFmpeg* даже используется YouTube на стороне сервера и в *Google Chrome* локально для работы с видео- и аудиоданными HTML5.

В новой версии множество улучшений, и теперь поддерживается протокол UDP Lite, позволяющий воспроизводить поврежденные в сети части через попытки восстановления утерянных данных локальным декодером *FFmpeg*. Кроме того, *FFmpeg* также начал поддерживать анимированные изображения WebP и APNG и многопоточность (благодаря слиянию с *ffmpeg-mt*), новые мультимплексы-демультиплексы, включая контейнер HEVC/H.265 RTP, а также поддержку фрагментированных потоков MPEG-DASH.

Эта функция означает, что видеопоток разделен на фрагменты определенного размера, и если пропускная способность сети сокращается, то плеер на основе *FFmpeg* может автоматически снизить битрейт, ухудшив качество, но не прервав воспроизведения. Кроме того, *FFmpeg* предлагает практически все улучшения *libav*, конкурирующего ответвления, и все это опять же благодаря солидной работе по поддержке обратной совместимости.

Конечно, это не единственный вариант поддержки мультимедиа в Linux, однако он может похвастаться высочайшим качеством кода. Проект анализирует все входящие патчи-заплаты с помощью собственного регрессивного тестового пакета FATE (аббревиатура от *FFmpeg*

«FFmpeg также предлагает почти все усовершенствования libav.»



Если вы хотите, чтобы ваш плеер использовал последний *FFmpeg*, позаботьтесь об установочном префиксе и укажите его в команде `./configure`.

Automated Testing Environment), и если созданный патч замедляет работу более чем на 0,1 %, он отвергается.

Сайт проекта предлагает очень подробное описание практически всех основных библиотек, и оттуда же можно загрузить исходный код. Для компиляции кода нужна некоторая подготовка, например, обзаведение стандартным *dev-stack* *Yast* для создания программ на вашем компьютере. Ряд компонентов *FFmpeg* опционален, и скрипт `./configure` сообщит вам, какие из них попадут в сборку. Новые функции и дополнения к серии 2.5.x являются обязательными.

Приложение Dropbox CLI

DB Uploader

Версия: 0.14 Сайт: <http://bit.ly/Dropbox-Uploader>

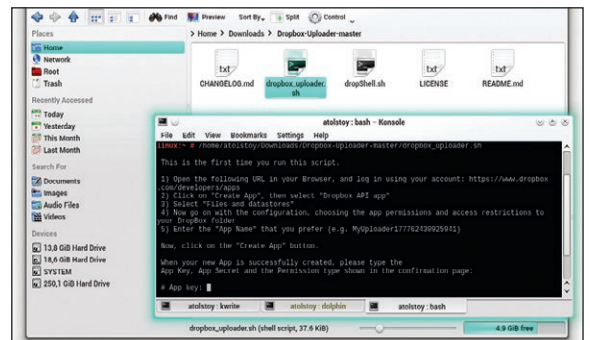
Не так давно API Dropbox изменился таким образом, что выполнять основные действия с файлами и папками из командной строки стало невозможно. Например, ради большей безопасности публичные ссылки (на файлы с распределенным доступом в папке *Public*) изменились так, что их стало возможно создавать только после переговоров с облаком, которое возвращало ссылку с уникальным хэш-значением, рассчитанным на стороне сервера.

На данный момент нет сервисного меню для *Dolphin*, *Thunar* и прочих менеджеров файлов. Конечно, пользователи могли бы создать эти меню самостоятельно, если бы могли общаться с облаком Dropbox напрямую. Благодаря Андреа Фабрици [[Andrea Fabrizio](#)] сейчас это стало возможно (снова) с помощью его великолепно *Dropbox Uploader*. Это скрипт *Bash*, который восстанавливает взаимодействие CLI с Dropbox, добавляя себя в качестве

стороннего приложения и соединяясь с сервисом.

Скачайте мастер-архив ZIP со страницы проекта Github, распакуйте его и запустите `./dropbox_uploader.sh`. При первом запуске скрипт проведет вас по процессу настройки. Здесь очень хорошая встроенная документация со всеми необходимыми подробностями. Скрипт объяснит шаги по регистрации скрипта как стороннее приложение, которому Dropbox предоставляет доступ для работы с общими файлами. Когда вы вставите вновь созданный ключ приложения и секретную подсказку в поле скрипта, начальная настройка завершится. Программа загрузки не требует дополнительной аутентификации

«Скрипт восстанавливает взаимодействие CLI с Dropbox.»



При первом запуске скрипта удобный мастер проведет вас по первичной настройке.

и по этой причине не хранит никаких особо важных данных. Она работает с официальным API Dropbox и позволяет копировать, перемещать, удалять и переименовывать файлы в вашей учетной записи, а также загружать, скачивать и выдавать доступ к файлам. Например, получить публичную ссылку на файл можно вот так просто:

```
./dropbox_uploader.sh share Public/My_file.odt
```

Папка `~/Dropbox` используется как корневая, полный путь предоставлять не нужно. Если файл существует, он синхронизируется, и вы получите действующую ссылку от Dropbox и уведомление об успешности действия. Скрипт `uploader` не зависит от официального клиента, так что он еще более кросс-платформенный.

Инструмент быстрого предпросмотра

Gloobus-preview

Версия: 0.4.5 Сайт: <http://bit.ly/GloobusPreview>

Э то хорошо известная программа быстрого предпросмотра файлов, изначально разработанная для сред рабочего стола на GTK, но подходящая для почти любого рабочего стола. Приложение поддерживает изображения, документы (PDF, ODF, ODS и т.д.), аудио (MP3, WAV и OGG), видео (AVI, OGG, MKV, FLV и т.д.), папки, архивы, шрифты, простые текстовые файлы и многое другое.

В Gnome, Unity или Cinnamon *Gloobus-preview* интегрируется в менеджер файлов *Nautilus* или *Nemo* (для предпросмотра нажмите пробел), а в KDE можно вручную создать простое меню сервиса, которое будет открывать файл для быстрого просмотра:

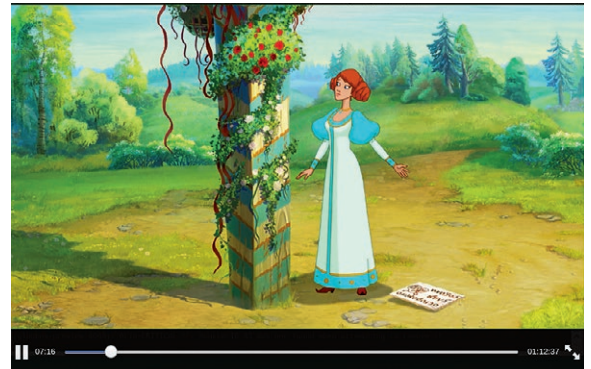
```
gloobus-preview %f.
```

Прошлая версия *Gloobus-preview* появилась в конце 2009 г., и с тех пор приложение утратило совместимость с современными дистрибутивами Linux. Однако недавно вступивший в проект новый разработчик Дьёрдь Балло [Gyorgy Ballo] внес

несколько основных обновлений в код. Он портировал пользовательский интерфейс в *Pygobject3* и *GTK3*; среду мультимедиа в *GStreamer 1.0*; и ключи мультимедиа в *GDBus*. Солидно расширился также список поддерживаемых форматов файлов. Старый движок *icns* сменился на *ImageMagick*, и добавилась поддержка формата XPS; улучшены общие возможности офисных плагинов с помощью *ssconvert* (развившегося из *Gnumeric*) и поддержка архивов благодаря *bsdtar*.

Gloobus-preview не одинок в своем классе: имеются различные проекты-конкуренты, от *Gnome Sushi* и *Nemo Preview* до *Klook*. Однако *Gloobus-preview* предлагает самый широкий список поддерживаемых типов файлов, выглядит

«Имеет самый широкий список форматов и работает гладко.»



Предпросмотр очень привлекательно выглядит в любой среде и работает как легковесная программа для чтения и плеер.

весьма отполированным продуктом и работает гладко.

Установка новой версии на Ubuntu и его производных весьма проста благодаря соответствующим PPA (<http://bit.ly/WebUpd8PPA>). На других системах вам, возможно, придется конвертировать Deb-пакет Ubuntu с помощью *Alien* (он отлично работает в openSUSE), или скомпилировать просмотрщик из исходника. В последнем случае проверьте, чтобы у вас была последняя версия (используйте URL в ленте), а не устаревшая 4.1.

Приложение Youtube CLI

MPS-Youtube

Версия: 0.2.1 Сайт: <http://bit.ly/MPS-Youtube>

С MPS-Youtube мы продолжаем нашу серию инструментов командной строки, упрощающих управление вашими любимыми приложениями и сервисами. *MPS-Youtube* вначале планировался как аудиоплеер и программа скачивания треков для YouTube, но очень скоро обзавёлся также и функцией воспроизведения видео.

Проект основан на MPS, программе терминала для поиска, стриминга и скачивания музыки. Название намекает, что *MPS* привязана к YouTube как источник музыки и видео, и в приложении также имеется достойная поддержка функций YouTube. *MPS-Youtube* умеет находить метаданные (счетчик просмотров, продолжительность, рейтинг, автор, эскиз и ключевые слова), создавать и сохранять локальные списки воспроизведения, искать и импортировать списки воспроизведения YouTube, просматривать комментарии для видео, и т.д. Все это стало возможным благодаря

библиотеке *Pafy* (<http://bit.ly/PafyLib>), которая уже включена в *MPS-Youtube*.

Приложение находится в пакетах в виде модуля Python (поддерживаются серии 2.7 и 3.x), и его легко установить из каталога *pip*. Проверьте наличие у вас в системе пакета *python-pip* (или с похожим названием) и введите следующее:

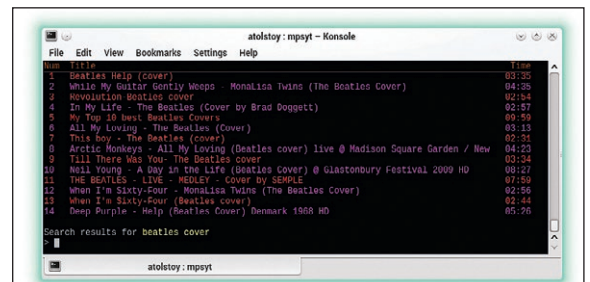
```
sudo pip install mps-youtube
```

После установки запустите его командой *mpsyf*. По умолчанию в *MPS-Youtube* не включено воспроизведение видео, так что давайте это исправим:

```
set show_video true
```

И еще одно необходимое условие для включения функции поиска по видео (по умолчанию ищется только музыка):

«Поиск, стриминг и воспроизведение музыки и видео.»



Это приложение командной строки способно заменить ваш браузер; оно предлагает и плагин Flash, и расширения YouTube.

```
set search_music false
```

Теперь вы можете осуществлять поиск по видео на YouTube, используя точку или слэш, за которыми следует переменная поиска. Например:

```
.beatles cover
```

Приложение вернет таблицу с результатами поиска в двух столбцах: номер и соответствующее название. Введите требуемый номер и нажмите Enter, чтобы начать просмотр. Вы можете изменить видеоплеер по умолчанию, *MPlayer*, на какой-нибудь другой, например, *MPV*, с помощью

```
set player mpv
```

Чтобы вернуть результаты поиска, закройте окно видео и нажмите в оболочке **Ctrl+C**. Вы можете легко скачать лучшее аудио или лучшее видео версии трека, используя соответственно **da #** или **dv #**.

Звуковой редактор

KWave

Версия: 0.8.99-2 Сайт: <http://bit.ly/KwaveApp>

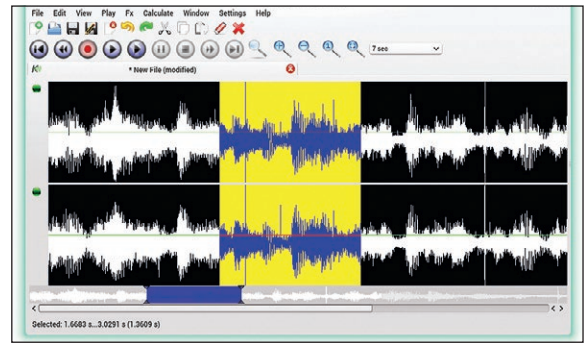
KWave — звуковой редактор, созданный на интерфейсе Qt4 и, следовательно, отлично сочетающийся с KDE и любой другой средой рабочего стола на Qt. Обработка данных звуковых волн в Linux считается побочным занятием, и здесь не хватает высококачественных редакторов. Самым мощным решением считается Audacity, но иногда бывает нужно более легковесное приложение, способное на нечто большее, чем устаревшая kde3-krecord. KWave вполне отвечает этим требованиям. Он появился в 2004 году, и в настоящее время обновляется раз в несколько месяцев.

KWave может работать с PulseAudio, ALSA и даже звуковой системой OSS, которую можно настроить в Settings > Playback. После этого начинайте запись, нажав на красную кнопку. Появится новое окно с настройками записи, где можно изменить качество звука, предварительно определить временной промежуток записи, устройство ввода и т. д.

Новая запись появится в новой вкладке в KWave. В меню FX вы также найдете такие важные эффекты, как Normalise, низкочастотный [Low] и полосовой [Band] фильтры, Pitch Shift [изменение высоты тона] и Notch filter [режекторный фильтр]. В меню Calculate вы можете добавить в свою запись шум или заглушить звук выбранной области.

График можно увеличить и уменьшить, или выделить какую-то его часть и экспортировать ее в файл. KWave поддерживает обработку нескольких треков со всеми функциями, необходимыми для создания сложной многодорожечной композиции. Затем вы можете экспортировать готовую работу в формате OGG, FLAC, WAV или даже MP3.

«KWave может работать с Pulseaudio, ALSA и даже OSS.»



Если вы ищете эквивалент Audacity для KDE, вы, вероятно, нашли оптимальный.

В новой версии исправлены многие прошлые проблемы со стабильностью, добавлены новые переводы и несколько новых аргументов командной строки. Например, можно быстро загрузить нужную запись, таким образом:

```
kwave --iconic --disable-splashscreen test.wav
```

KWave также поддерживает удаленное использование через такие методы, как `kwave:plugin %3Aexecute?normalize, kwave:save`

KWave включен в директорию приложений Ubuntu, хотя иногда он там не обновлен до самой свежей версии. Официальная страница KWave (см. выше) предлагает последние версии для большинства RPM-дистрибутивов и исходный код.

Менеджер пакетов

Linuxbrew

Версия: rolling dev Сайт: <http://brew.sh/linuxbrew>

Все, кто пробовал Mac OS X, знакомы с системой пакетов Homebrew. Для пользователей Apple управление пакетами — не совсем обычное дело, а в Linux это основополагающая функция. Однако Homebrew сильно отличается и от Macports (стиль BSD), и от Apt, Yum и т. п. для Linux.

Linuxbrew — это ответвление Homebrew для Linux, практически с теми же функциями, что и его родитель, так что давайте пройдемся по самому основному.

Linuxbrew — скрипт Ruby, который выдает доступ к онлайн-репозиториям программ, в частности, системных инструментов, пакетов Ruby on Rails и соответствующих дополнений. Если вы не являетесь разработчиком Ruby, перед вами откроется целый мир новых программ на <http://braumeister.org>.

Чтобы установить клиент Linuxbrew, сначала надо проверить наличие у себя обычного арсенала программиста (`curl`,

`git`, `m4`, `ruby`, `texinfo`, `bzip2-devel`, `curl-devel`, `expat-devel`, `ncurses-devel` и `zlib-devel`). Поготовившись, введите команду установки —

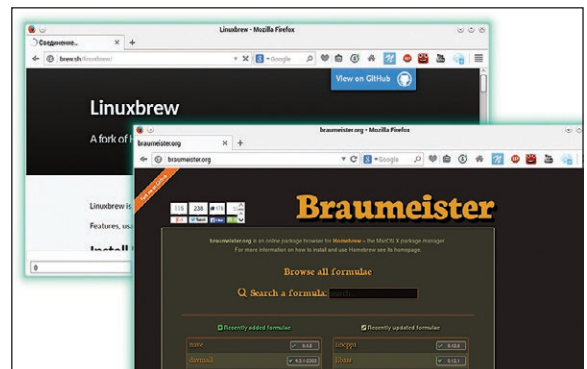
```
git clone https://github.com/Homebrew/linuxbrew.git ~/.linuxbrew
```

и добавьте нижеследующие строки в файл `.bashrc`:

```
export PATH="$HOME/.linuxbrew/bin:$PATH"
export MANPATH="$HOME/.linuxbrew/share/man:$MANPATH"
export INFOPATH="$HOME/.linuxbrew/share/info:$INFOPATH"
```

После этого установка любого пакета сводится всего лишь к `brew install package_name`

«Если вы не разработчик Ruby, вам откроется целый мир программ.»



Просмотрите репозитории онлайн или используйте команду поиска brew, чтобы найти нужный пакет.

Linuxbrew не нужна аутентификация root или sudo, поскольку он размещает установленный файл в вашей домашней папке под `~/.linuxbrew`. Решение Linuxbrew может показаться не слишком уместным в старых дистрибутивах LTS Linux, которые больше не предлагают свежих версий инструментов разработки. Однако этот выбор намного богаче, чем просто склад разработчика. Homebrew включает множество репозитория, в том числе множество популярных игр (`Freeciv`, `SuperTux` и т. д.). База данных приложений для Homebrew и Linuxbrew одна и та же, так что при необходимости синхронизация вашего пакета разработки между OS X и Linux не составит труда.

HotGames Развлекательные приложения

Имитатор гонок

SuperTuxKart

Версия: 0.8.2b Сайт: <http://bit.ly/SuperTK>

SuperTuxKart — это симулятор гонок на картах, который появился в 2004 г. как свободный и открытый клон *Mario Kart* от Nintendo. С тех пор игра обрела новые функции и технологии, а также обзавелась симпатичными подробностями из многих проприетарных симуляторов картинга, в частности, *Moorhuhn Kart*. В игре участвуют популярные талисманы из мира открытого кода, в том числе родной Tux, демоненок Beastie из BSD, Wilber из GIMP, голубь Pidgin, дракончик Konqi и Крыс XFCE. Каждого из них можно сделать водителем, а прочие будут вашими соперниками. Кстати, судьей является Thunderbird.

Последняя стабильная версия SuperTuxKart — 0.8.1, она относится к ноябрю 2013 г., и вряд ли был бы особый смысл писать о ней в HotPicks, если бы в последнее время в репозитории Git игры

не появились важные изменения. И они реально новаторские.

Самая свежая нестабильная версия 0.8.2b использует новый графический движок Antarctica, который поддерживает 100 % динамическую светотень, объемный туман, глубину поля, разные модели карт (glossmap/specular map/normal map/emit map) и требует совместимого с OpenGL 3.1 видеочипа, чтобы можно было воспользоваться всеми этими прелестями. Есть также новый движок рендеринга на основе шейдера, физика, гоночные трассы, поддержка Wii, поддержка учетных записей онлайн

«Новый движок рендеринга на шейдере, физика и трассы.»



Этот любимый симулятор картинга с открытым кодом вернулся с новыми силами и здорово улучшенной графикой.

и долгожданный многопользовательский режим.

Настройки игры позволяют менять качество графики, но помните, что для использования самых высокоскоростных настроек (с динамической подсветкой и т. д.) нужна соответствующая графическая карта. Официальным минимальным требованием является Intel HD3000, но наилучшие результаты вы получите при наличии высококачественных карт Nvidia или AMD, с проприетарными графическими драйверами.

Настольная игра

Aualé

Версия: 1.0 Сайт: www.joansala.com/auale/

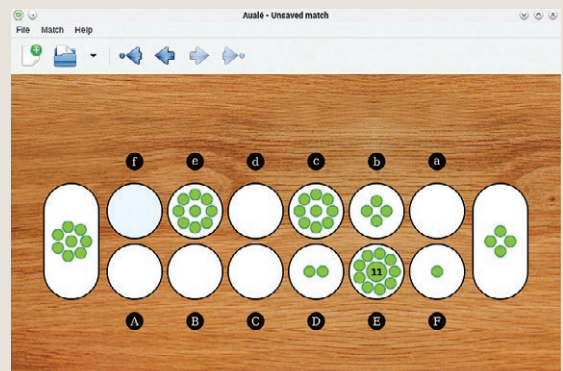
Aualé — стратегическая настольная игра с долгой историей. Ее название — перевод на каталонский от Mancala, названия семейства настольных игр из Африки и Среднего Востока. Основной принцип — перенести семена из одной лунки на вашей стороне поля и затем распределить их (засеять) в лунки против часовой стрелки, по семечку на лунку. При определенных условиях можно забирать семена из лунок ваших соперников в свой амбар.

В *Aualé* каждый игрок начинает с шести лунок, и всего имеется 48 игровых фишек или семян. По правилам вы можете брать семена для своего амбара, когда в лунке вашего соперника имеется два или три семечка. Игрок, набравший 24 семечка первым, выигрывает — и *Aualé* оказывается игрой довольно сложной, даже если играть против компьютера (AI), который предлагает

только базовый уровень искусственного интеллекта.

Игра требует постоянной переоценки рисков и возможностей, и, конечно, точного подсчета количества семян в каждой лунке. Оба игрока стремятся добиться того, чтобы у соперника вообще не осталось семян на его стороне и, следовательно, места для маневра. В таком случае весьма легко дожать соперника и выиграть у него. Естественно, долгая игра требует хорошо продуманных позиционно ходов, когда игроки сосредотачивают семена в нескольких лунках и очень аккуратно и тщательно их распространяют.

«Aualé оказывается сложной, даже если играть против AI.»



Взвинтившись на гонках, мы провели пару часов с семенами и лунками. И успокоились.

Aualé позволяет потягаться с компьютером. Вы также можете сохранить игру и продолжить ее позже, а также отмотать назад и вперед недавно сделанные ходы. Сайт игры предлагает бинарники для многих популярных дистрибутивов Linux, на основе Deb и RPM. Однако начинка пакета показывает, что *Aualé* состоит из скриптов и привязок Python и независимых от платформы частей JAR. Если вы любите логические игры, то *Aualé* — достойный выбор.

Имитатор деятельности

Hollywood

Версия: 1.8 Сайт: <http://bit.ly/HWTechMelodrama>

Есть редкая категория приложений, которая развлекает, доставляя чистую радость, но на практике чуть более чем никчемная. Дастин Керклэнд [Dustin Kirkland] и Кейз Кук [Case Cook] из Canonical потратили свой досуг на создание совершенно бесполезного, но и блистательного пакета для консоли Ubuntu: он называется *Hollywood Technical Melodrama*.

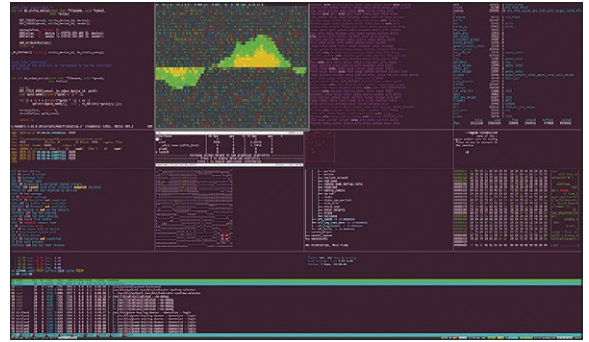
И что же это? Этот пакет превращает вашу консоль Ubuntu в интерфейс хакера из голливудской технической мелодрамы, соединяя «хакерство реальной жизни», как оно показано в классических фильмах типа «Хакеры» 1995 г., и бегущий текст и графику, которые вы обычно видите на больших экранах каждый раз, когда нужно показать какой-нибудь жутко секретный центр управления в боевиках. Мы сильно сомневаемся, что АНБ или штаб правительственной связи будут раздвигать тяжелые мраморные панели, чтобы взглянуть на огромные карты, начиная

каждую новую миссию (кошмар и для злодеев, и для безопасности); естественно, и на большом экране подобные сцены частично или даже полностью бессмысленны, но до этого никому нет дела, если фильму удастся создать атмосферу зловещего бункера или затей.

Пакет *Hollywood* поддерживает все недавние версии, начиная с 12.04, а также все прочие, совместимые с разновидностями и производными *buntu. Если у вас не Ubuntu, вы все равно можете скачать пакет [tar.gz](http://bit.ly/1Aig3x2) вручную (<http://bit.ly/1Aig3x2>), распаковать его и запустить.

Приложение не содержит бинарников, но имеет немало зависимостей; некоторые из них — знакомые инструменты командной строки. *Hollywood Technical Melodrama*

«Доставляет чистую радость, но на практике никчемная.»



Этот скриншот зарезервирован для фильма о разработке Mir.

использует *Yubuntu*, текстовый менеджер окон, который создает сессию `tmux` и делит основное окно на части. И затем в них отображаются результаты разных команд. В качестве бонуса, скрипт создает несколько раз в минуту изменяет структуру *Yubuntu*. Чтобы *Hollywood* запускался гладко, понадобятся *htop*, *MPlayer*, *mlocate*, *ccze* и ряд других утилит. Если чего-то не хватает, просто просмотрите записи об ошибках и жалобы. Полнофункциональный *Hollywood* должен запускать цветной текст в стиле Матрицы, логи и даже файл MP4, воспроизводимый видеовыводом ASCII. Конечно, нагрузка на CPU будет весьма высока, так что следите за уровнем заряда батареи, когда будете демонстрировать всем и каждому в кофейне свои крутые хакерские навски.

Виртуальный принтер

Boomaga

Версия: 0.6.2 Сайт: www.boomaga.org

Эта программа — огромное облегчение для всех, кто когда-либо хотел напечатать буклет в Linux. Большинство текстовых редакторов и графических приложений Linux обычно дают стандартные настройки печати с выбором страницы и доступом к опциям драйвера принтера, и этого более чем хватает, если только перед вами не стоит особая задача.

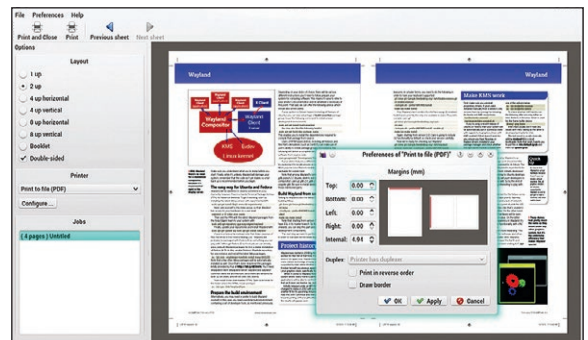
Например, может понадобиться разместить две страницы на одном листе и создать сшитый внакидку двойной буклет, где на первом листе — первая и последняя страницы, на следующем — вторая и предпоследняя, и т.д. И все это должно быть распечатано с двух сторон. В былые дни только *PageMaker* (и потом *InDesign*) предлагали плагин *Booklet*. Иногда эта функция включалась в драйвер принтеров некоторых моделей.

Boomaga (BOOklet MAnager) изящно справляется с этой задачей. При запуске его как отдельное приложение придется

заранее подготовить свой файл PDF, поскольку это единственный поддерживаемый формат файлов. Однако *Boomaga* заодно добавляется в список ваших принтеров, и его можно использовать из любого приложения с поддержкой печати.

После того, как вы выбрали распечатку в *Boomaga*, программа автоматически сгенерирует PDF и отобразит его в главном окне *Boomaga*. Здесь можно сортировать страницы так, как вам хочется разместить их на бумаге. Раскладка бывает в один блок (простая раскладка), два, четыре и восемь блоков или в виде буклета (для сшивания внакидку после того, как страницы будут перевернуты пополам). Назначив свой вариант раскладки, выбирайте

«Облегчение для всех, кто хочет распечатать буклет в Linux.»



Разместите страницы по-другому и распечатайте книгу, как истый профессионал.

настоящий принтер и доступ к его опциям, нажав на кнопку *Configure*. Опции бывают разными, в зависимости от вашей модели. Если в вашем принтере нет опции дуплексной печати, вероятно, нужно будет включить в настройках *Boomaga* *Layout*. Тогда приложение попросит вас перевернуть страницы посреди процесса печати вашего документа. Приложения размещают все задачи печати в собственной очереди, позволяя экспортировать каждую задачу в формате PDF (очень удобно для предварительной печати, переименования задачи, поворота страницы и т.д.)

Официальная страница скачивания *Boomaga* дает инструкции по установке. К счастью, есть пакеты для большинства популярных дистрибутивов Linux. LXF

На диске

Дистрибутивы, приложения, игры, подкасты и всякое-разное...

Лучшее из Интернета, упакованное в 9 ГБ качественного DVD.



Дистрибутивы

После установки каждого из дистрибутивов на DVD этого месяца стала очевидна разница между их инсталляторами. В Ubuntu простая и понятная графическая программа. Fedora, самый опытный и старейший дистрибутив в этой области, предлагает довольно приятную, но иногда путаную *Anaconda*, а ArchBang открывает окно терминала, хотя и вполне симпатичное, и задает вам вопросы в виде простого текста. Сначала подход ArchBang кажется мудреным и недружелюбным к тем, кто предпочитает GUI, однако у него есть важные преимущества. Все стадии процесса установки представлены в главном меню, вы можете перейти к нужной вам стадии, просто нажав на клавишу — а попробуйте сделать так в GUI. Вы сразу видите, далеко ли вы продвинулись в ритуале ответов на вопросы в процессе установки, и она не будет ничего делать, пока вы не ответите на все вопросы и не скамандуете продолжить. Это чуть медленнее, но зато гораздо удобнее.

В стремлении сделать все как можно дружелюбнее к пользователю с помощью милого GUI типа наведи-и-щелкни, не пошли ли мы неверным путем, пожертвовав функциональностью? Дружелюбие — дело хорошее; но отупление — нет.

Neil

» 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.

Дистрибутив пользователя среднего уровня

64-битный

Fedora & Ubuntu

На LXF DVD прошлого месяца у нас был релиз Fedora 21, с несколькими дополнительными рабочими столами. Однако когда мы подробно рассмотрели последние релизы Fedora

(на данный момент имеется три разновидности: Cloud, Server и Workstation) в одном из наших материалов в этом выпуске (стр. 52), стало ясно, что есть смысл снова его включить, чтобы вы могли попробовать новые функции, о которых рассказал Джонни.

Однако помните, что на сей раз там есть только стандартный рабочий стол Gnome 3 (или, если точнее, 3.14). Как это часто бывает, после выхода релиза появилось множество обновлений, и чем больше людей им пользовалось, тем больше выявлялось и исправлялось ошибок. Поэтому в этом месяце в предложение от Fedora также входит ре-спин, на сей раз со всеми обновлениями за шесть недель после релиза. Ubuntu 14.10 на LXF DVD тоже обновлен.



Дружелюбный к новичкам дистрибутив

32-битный

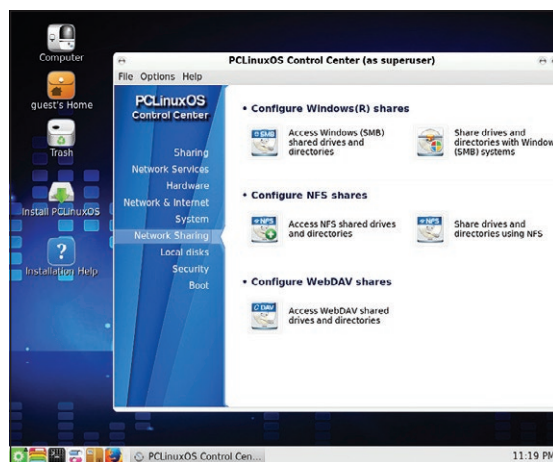
PCLinuxOS 2014.1

Во мраке отдаленного прошлого, когда Ubuntu еще не был популярнейшим в мире дистрибутивом Linux, звездой был Mandrake. Слияние с Connectiva и затем ответвление в Mageia достаточно хорошо известны, но была и более ранняя ветка Mandrake, под названием PCLinuxOS. Он появился в качестве дополнительного программного репозитория для Mandrake — вероятно, основного источника неофициальных пакетов — и постепенно вырос в самостоятельный дистрибутив.

Он был, и остается, простым в использовании дистрибутивом, идеально подходящим для начинающих пользователей Linux. И именно этот дистрибутив ввел идею live CD, который можно использовать для демонстрации дистрибутива (или его использования) с последующей установкой в систему, если пользователь остался им доволен.

Однако следует помнить, что современная загрузка PCLinuxOS не особенно хорошо

сочетается с устройствами USB, будь то флешка или внешний диск DVD. Вам придется загружаться с внутреннего DVD привода, или используя LXF DVD, или записав образ PCLinuxOS ISO на DVD.





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.

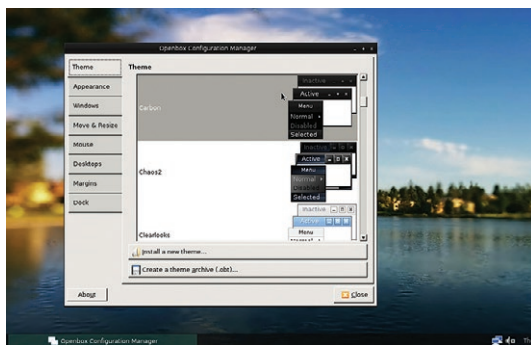
Легковесный дистрибутив

32-БИТНЫЙ

ArchBang 2015.01

АrchBang — еще один легковесный дистрибутив, черпающий вдохновение в превосходном #!CrunchBang Linux. А судя по другой части названия, это один из множества дистрибутивов,

принявших за основу Arch Linux. Если вам нравится работа с live CD и вы хотите установить ArchBang, подготовьтесь. Установщик отнюдь не является дружелюбным милашкой, ведомым мышью. Эта текстовая программа задает вам вопросы, и здесь нужен некий уровень знания своей системы. Но она не тронет ваш диск, не получив все ответы, так что это вполне безопасно.



Решив опробовать тестовую установку, мы выбрали в качестве загрузчика *Grub*, и установленная система повисла, даже не загрузившись. Попробовав снова и на сей раз выбрав *Syslinux*, мы получили систему без всяких проблем.

Легковесные дистрибутивы часто считают идеальными для старого «железа», а уж на новом ПК они будут просто летать, оставляя большую часть его ресурсов для настоящей работы.

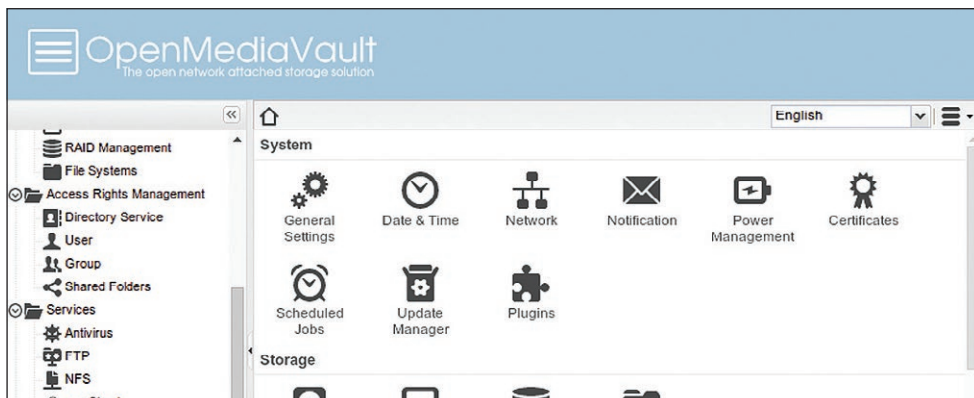
Дистрибутив NAS

64-БИТНЫЙ

OpenMediaVault

Этот дистрибутив довольно-таки специфический, однако он идеально подойдет для того замечательного компьютера, который пылится у вас с тех самых пор, как вы купили себе новый на Рождество. OpenMediaVault 1.9 превращает компьютер в машину NAS (Networked Attached Storage).

Это не дистрибутив live — его нужно устанавливать на жесткий диск, так что скопируйте ISO-образ с DVD на CD или устройство USB и загрузитесь с него. После установки он предоставит вам URL, который вы сможете использовать для администрирования и использования NAS.

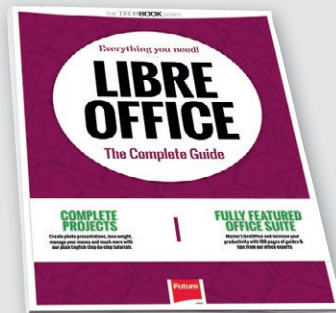


И еще!

Бесплатная е-книга LibreOffice: Полный гид

Все необходимое для освоения работы в мощном офисном пакете *LibreOffice* — по невероятной цене, то есть вообще бесплатно. Выгодно! В этом месяце *LXF DVD* включает е-книгу из нашей серии TechBook, которая в обычных условиях стоила бы вам £5,99, и в ней вы найдете 100 страниц руководства и советов, написанных на отличном английском языке нашими экспертами.

Руководство охватывает все приложения в пакете. Например, вы научитесь создавать стильные рекламные проспекты в *Writer*, использовать главные документы и вставлять информационные заготовки в письма. В *Calc* & *Math* мы покажем, как управлять своими финансами, делить громоздкие электронные таблицы на части и упрощать управление сложными таблицами с помощью сводных таблиц. В *Impress* вы научитесь создавать потрясающие презентации, а в *Draw* мы погрузимся в мир графиков и логотипов. *LibreOffice* содержит даже приложение базы данных, и мы поможем вам разобраться в том, как создать базу данных и как использовать формы и запросы.



Системные инструменты

Главное

Checkinstall Установка tar-архива с помощью менеджера пакетов.

GNU Core Utils Основные утилиты, обязанные быть в каждой ОС.

Hardinfo Инструмент для тестирования ОС.

Ядро Исходник последней версии ядра.

Mentest86+ Проверьте свое ОЗУ на сбои.

Plop Простой менеджер запуска ОС.

RaWrite Создавайте загрузочные дискиеты в Windows.

SBM Независимый от ОС менеджер загрузки с простым интерфейсом.

VvDial Соединяйтесь с Интернетом через телефонный модем.

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:




LXF191/192
Январь 2015

250 руб.

- » Пиршество идей Проекты для Raspberry Pi
- » Звуки музыки Плееры для фонов
- » Робот — друг человека Соберем для дома, для семьи
- » Забыли пароль? Rescatux вас спасет

LXFDVD: все для Raspberry Pi — Raspbian, KaliLinux, PiMusicBox, RetroPie, Jasper, NOOBS; 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_191-192/
PDF-версия: shop.linuxformat.ru/elxf_191-192/




LXF193
Февраль 2015

250 руб.

- » Вещание по дому ПК с Linux как хаб медиа-файлов
- » Векторная графика Объедем контуры на кривых Безье
- » Отстрел диска Не доставайся же ты никому!
- » Горячие точки Визуализация статистики

LXFDVD: OpenELEC 4.95.3, openSUSE 13.2, Ubuntu 14.0, XBMCubuntu 13.2, HandyLinux, Netrunner, Robolinux, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_193/
PDF-версия: shop.linuxformat.ru/elxf_193/



LXF194
Март 2015

250 руб.

- » Круче — только... Технологии-2015
- » Стол как предмет выбора Понятное дело, рабочий
- » 3D на Pi Недетская графика на компьютере-крошке
- » Враг не пройдет Тестируемся на вторжения

LXFDVD: 4MLinux, ALT Linux, Fedora, Manjaro, Bodhi, LMDE 2 Cinnamon, Quirky April 7, Tails, Voyager X, справочник по GIMP и 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_194/
PDF-версия: shop.linuxformat.ru/elxf_194/

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

» Содержание

ЛINUX
ФОРМАТ

ДИСТРИБУТИВЫ

Fedora 21 (64-битный)
Отлично отлаженный и простая в использовании ОС для ноутбуков и настольных ПК с полным инструментарием и помощниками для разработчиков и умельцев.

Ubuntu 14.10 (64-битный)
Новейший Ubuntu с добавленной ценностью от нас: выбор из пяти рабочих столов, переводы — включая Malay.

PC Linux OS 2014.12 (32-битный)
Простая в использовании ОС для не слишком новых ноутбуков и настольных ПК.

ArchBang 2015.01
Быстрый, стабильный и легкий дистрибутив со скрывающимся релизом (всегда обновлен), на базе оконного менеджера Openbox Window Manager. Подходит и для настольных, и для портативных ПК.

СРАВНЕНИЕ

Языки программирования
Bash, perl, C#
Perl
Python
Ruby

ПОТРICKS

Далеко Настольная плата
Воюющая Виртуальный принтер
Droidbox Ubuntu Приложение для Android
FTP-ред Набор мультимедиа-библиотек и программ

GlobalS-Printer Инструмент быстрого предпросмотра
Налиwood Симулятор деятельности Клиента Звуковой редактор

LinuxDev Менеджер пакетов
MP3-YouTube Приложение YouTube с SLI
SuperTuxKart Оригинальный симулятор
Yaseok Музыкальный плеер

УЧЕБНИКИ: ФАЙЛЫ ДЛЯ УРОКОВ

MS-DOS, Rak
Uala
Motlon
Python

ПОМОЩЬ

Руководство новинка
Руководства
Ответы

Окончание на обороте »

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!

ЛINUX ФОРМАТ
5 ПОЛНЫХ ДИСТРИБУТИВОВ И БОЛЕЕ ТОГО!
ВСЕ ДЛЯ БЫСТРОГО СТАРТА В LINUX

ФЕДОРА 21
РАБОЧАЯ СТАНЦИЯ
Главные
авангардные
функции Red Hat

СО СВЕЖИМИ
ИСПРАВЛЕНИЯМИ

е-КНИГА ДАРОМ!
Libre Office
LIBRE OFFICE
The Complete Guide

64-битный
Ubuntu 14.10
Уторіс Unіcorn
Идеальный дистрибутив
Linux для новичков.
Установите или запустите
с нашего DVD прямо сейчас!

СО СВЕЖИМИ
ИСПРАВЛЕНИЯМИ

Содержание

LINUX FORMAT

ГЛАВНОЕ

- Checkinstall
- Coreutils
- HardInfo
- Kernel
- Memtest86+
- Plop
- SEMI
- Wvdial

ДОКУМЕНТАЦИЯ — 10 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

- Bash Scripting** Подробное руководство по программированию на Bash
- Bourne Shell Scripting** Начальное руководство по программированию на Bash

Cathedral Bazaar Классический текст Эрика Раймонда [Eric S Raymond] «Собор и базар»

The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian

Dive Into Python Учебник по программированию на Python

Intro to Linux Начальное руководство по Linux

Linux Dictionary Словарь Linux, объясняющий специфическую терминологию

Linux Kernel in a Nutshell Описание ядра Linux, созданное одним из его выдающихся разработчиков — Грегом Кроа-Хартманом [Greg Kroah-Hartman]

System Administrators Guide Руководство по базовому администрированию Linux

GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

ДИСТРИБУТИВЫ

OpenMediaVault 1.0.9 (64-битная версия) Решение для сетевого хранилища (NAS) нового поколения на базе Debian Linux. Содержит такие сервисы, как SSH, (S)FTP, SMB/CIFS, медиа-сервер DAAP, RSync, клиент BitTorrent и многое другое. Благодаря модульной структуре, легко расширяется плагинами.

EXTX 15.2 LXQt (64-битная версия) Легковесный, модульный и поразительно быстрый дистрибутив, основанный на Debian Jessie/Ubuntu 15.04, с рабочими окружениями LXQt 0.9.0 и KDE 4.14.6 вместе с KDE Frameworks 5.9.0.

Дружелюбен к пользователю.

MauiLinux 8.0 "LxFce" (64-битная версия) Объединяет элементы рабочих столов LXDE и Xfce: благодаря движку LXDE работает гладко даже при небольшом объеме ОЗУ, а интерфейс Xfce обеспечивает большее дружелюбие к пользователю и лучшую управляемость.

Sailetus 14.04.2 (64-битная версия) Дистрибутив на базе Ubuntu, использующий в качестве оконного менеджера Openbox.

Задуман как легковесный, но тем не менее полнофункциональный и завершённый. Включает некоторые свойства Gnome и Xfce.

**Пожадуйста, перестаньте использовать
именем Аджного Диска озанковать
с инструкцией, опубликованной
в журнале на стр. 109!**

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕЯТЕЛЬНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данной диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аплиментарный 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. Представленные herein программы или данные. Прежде чем устанавливать какие-либо ПО на компьютер, пожалуйста, внимательно прочтите инструкцию к каждому из них.

Тираж изготовлен ООО «Марком», 188652, Россия, Ленинградская область, Всеволожский р-н, дер. Юрки, Школьная ул., 7-а. Лицензия ИМПР. ВАР N 77-03.

Создание установочных дисков при помощи cdcrcord

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/>

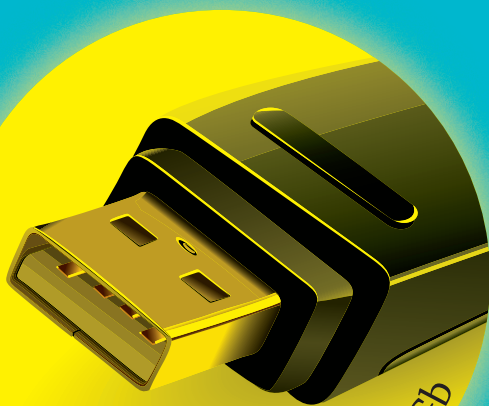
Отдел дистрибьюции ГНУ/Линуксцентра приглашает дилеров и дистрибьюторов к сотрудничеству!

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/

RH124, RH134, RH254
февраль 2015



RHEL7 в формате LVC

Пройди путь от новичка до инженера Red Hat не выходя из дома

**LVC (LIVE VIRTUAL CLASS) -
ИНТЕРАКТИВНОЕ ДИСТАНЦИОННОЕ ОБУЧЕНИЕ
В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ**

RH124 - RED HAT SYSTEM ADMINISTRATION I	02.02.15 - 06.02.15
RH134 - RED HAT SYSTEM ADMINISTRATION II	09.02.15 - 12.02.15
RH254 - RED HAT SYSTEM ADMINISTRATION III	16.02.15 - 19.02.15

WWW.UNIXEDU.RU

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/



В майском номере

100 самых лучших инструментов!

Не пройдите мимо лучших инструментов, утилит и приложений с открытым кодом, известных человечеству... в смысле, команде LXF.

Ящик с Tor

Мы (наконец-то) добрались до того, как СПО и немного электроники сделают анонимным ваш web-серфинг.

Храним инкогнито онлайн

Хит-парад лучших дистрибутивов-анонимайзеров... прямо и не знаем, кто их написал.

Домашний роутер — сами

Берем старую машину и создаем крутейший роутер, под руководством безумного гения Джонни.

Содержание будущих выпусков может меняться — кстати, я узнаю этот Tor из прошлого месяца...

LINUX FORMAT

Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № ФС77-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж печатной версии 2000 экз., распространение электронной версии 30000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Максим Алибаев, Елена Ессяк, Даниил Кривошеин, Светлана Кривошеина, Валентин Развозжаев, Елена Толстякова

Редактор диска

Александр Баракин

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Андрей Смирнов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный редактор

Павел Фролов

Учредители

Частные лица

Издатель

ООО «Линукс Формат»

Отпечатано в типографии ООО «ЛД-ПРИНТ»

196644, Санкт-Петербург, Колпинский р-н, пос. Саперный, территория предприятия «Балтика», д. 6/н, лит. Ф
Тел. (812) 462-8383, e-mail: office@ldprint.ru
Заказ 12442

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор [Neil Mohr] neil.mohr@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], Давид Эйтельбах [David Eitelbach], Кент Ельчук [Kent Elchuk], Мэтт Хансон [Matt Hanson], Расс Питт [Russ Pitt], Лес Паундер [Les Pounder], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Ричард Смедли [Richard Smedley], Александр Толстой [Alexander Tolstoy], Михалис Цукалос [Mihalis Tsoukalos], Стивен Ву [Steven Wu], Евгений Балдин, Михаил Вознесенский, Андрей Гондаренков, Петр Семилетов, Алексей Федорчук

Иллюстрации Шейн Коллиндж [Shane Collinge]

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 эксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных 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 ответственности за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds]. "GNU/Linux" заменяется на "Linux" в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт www.futureplc.com.



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

16+

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.

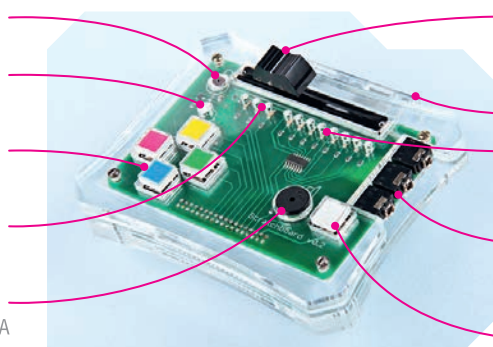
ДАТЧИК ЗВУКА*

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 **9500 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

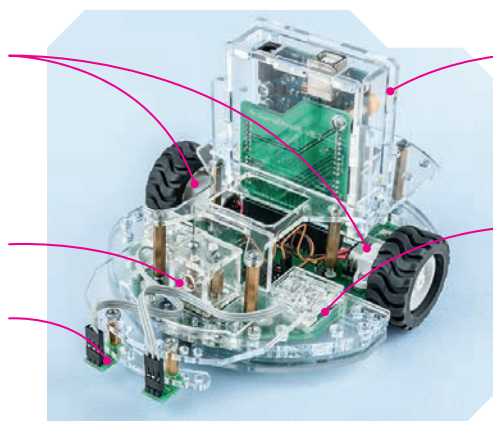
обеспечивает возможность:

- 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 **19 500 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)

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
- 30 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
- 50 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



**MADE IN
GERMANY**



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 30 ТБ/месяц (PX60), 50 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.