

DVD ДАРОМ!



НОВЫЙ UBUNTU 15.04!

LINUX FORMAT

Главное в мире Linux

Июль 2015 № 7 (198)

74 страницы учебников и обзоров!

- » Открываем SystemTap
- » Взламываем роутер
- » Строим сервер на Pi 2

+ Рассматриваем отличный Dell XPS 13

UBUNTU 15.04

Путеводитель по свежей версии

Подробный, все новинки поясняются

- » Systemd » Ядро 3.19 » Unity 7.3
- » Главные новые приложения
- » Версия с Mate

ЕСТЬ ЭЛЕКТРОННАЯ ВЕРСИЯ ДЛЯ ВАШЕГО ПЛАНШЕТА!



Платим вперед

« Делая открытый проект, чувствуешь, что все на равных правах. »

PayPal — про смыкание с открытым кодом **с. 44**

Безопасный Linux

» Как избежать нарастающих угроз от вирусов и эксплойтов



Системное программирование

Ядро сообщает

» Код для отслеживания изменений в файловой системе



Сравнение

Работа на расстоянии

» Рассматриваем самые быстрые и надежные решения



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





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.



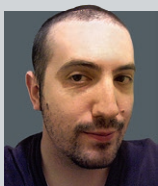
Кто мы

В этом номере мы спросили наших экспертов: что, по-вашему, Canonical должна бы добавить в Ubuntu?



Джонни Бидвелл

Я желаю, чтобы Ubuntu снова побурел. И выкинул Unity. Если вам нужен рабочий стол с причудами, который воображает себя наполовину на мобильном устройстве, так уж лучше GNOME 3. Да, и пусть вернут облачное хранилище Ubuntu One, как его ни назови — Ubuntu Two, UbuntuTutu или Ubuntu Zero.



Мэтт Билби

Вот за что я уважаю Canonical: с каждой новой итерацией интерфейс Unity хоть немного, да улучшается. Но релиз, которого я жду не дождусь — тот, что даст мне понять: уважение взаимно. Даже после всех этих лет я тоскую по уютной неэффективности горизонтального запускателя программ.



Лес Паундер

Был бы рад версии Ubuntu для сообщества самодельных, с Arduino, 3D-принтерами, ПО для прототипирования и т.д., установленными как стандарт и готовыми к моим опытам. Это приобщит многих к замечательной культуре созидания, разрастающейся по мере снижения барьера вхождения.



Маянк Шарма

Ubuntu One! Конечно, облачные сервисы уже идут по пятаку за пучок, но с сервисом Canonical я типа сросся. По-моему, как раз к моменту схлопывания он чудесно интегрировался с рабочим столом. Его мобильное приложение было милым и предлагало из магазина музыку без DRM, а уж это явный плюс.



Шашанк Шарма

Ратую за безупречную активацию голосом, а заодно уж приставить голову, все конечности и веселое настроение. Вышла бы отличная автоматизация. Но — чтобы маньяки клавиатуры сохранили возможность копаться в командной строке, настраивая свои личные процедуры и чертыхаясь.



Валентин Синицын

Хорошо бы иметь KDE «гражданином №1», как в SUSE. Да, я понимаю, что есть Kubuntu, но все же стратегическая линия — продвигать Unity. Если бы не писать все новое с нуля, а облагородить то, что есть — глядишь, и ракету в космос уже бы запустили.



Угрозы и уязвимости

» Азбука информационной безопасности гласит: риск возникает только тогда, когда источник угрозы и соответствующая уязвимость системы с некоей вероятностью «находят друг друга». С первым все более-менее ясно. Им может быть нарушитель, действующий осознанно и целенаправленно; обычный пользователь, «накосячивший» по незнанию или разгильдяйству; а то и вообще не человек — скажем, пожар в серверной. Вероятности появления каждого из них можно оценить более-менее точно. С уязвимостями всё сложнее. Их много, и с появлением новых технологий становится все больше. Такова плата за прогресс. Можно говорить о трех причинах возникновения уязвимостей. Первая — ошибки разработчиков. Примеры, связанные с недостаточной проверкой ввода в web-приложениях, вы найдете на страницах этого номера. Вторая — ошибки админов и безопасников: например, задержка установки обновлений или нечетко написанные инструкции для пользователей. Сюда же относятся и ошибки руководителей: не придали значения, не проследили, не организовали обучение и т.д. Третья причина — технические ограничения системы, делающие возможными атаки «отказ в обслуживании». С первой и последней причинами нам ничего не поделать (разве что попробовать оценить их вероятности на этапе выбора решения), а исключить вторую — вполне по силам. Неуязвимых систем не бывает, но риски уменьшить можно и должно. Главное — без паники...

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/linuxformat

» 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 гг.
- 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

«Ощутите свободу, которую предлагает Linux». Леннарт Пёттеринг

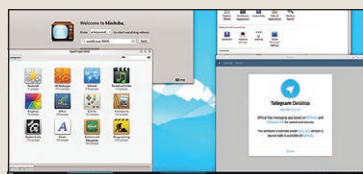
Обзоры

Bluestar Linux 3.0 16

Любитесь KDE-дистрибутивом на базе Arch: набит приложениями и использует *Pacman*.

SuperX 3.0 17

Такой ли суперский этот очередной KDE-дистрибутив? Как-то не похоже, что добавление условий пользования сильно помогает.



➤ Это дистрибутив, родства не помнящий; но он старается быть дружелюбным.

NethServer 6.6 18

Готовый к употреблению серверный дистрибутив Linux, способный упростить жизнь всем.

Unity 5 19

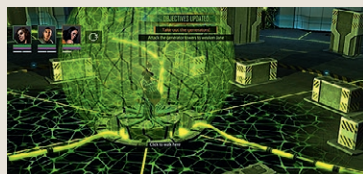
Ваша мечта стать разработчиком игр почти сбылась: движок — любимец Indie отныне поддерживает платформу Linux.

Blend4Web 20

Создавайте трехмерные презентации и выкладывайте их в Web! Средства позволяют.

Shadowrun Chronicles 22

Игра нетривиальная — а удачи ей добиться не удалось. В чем тут причина?



➤ Вначале главный герой понимает в происходящем не больше нас с вами.

Dell XPS 13 Dev Ed 24

Ноутбук Dell с Ubuntu на борту! Ваши дружки с MacBook'ами позавидуют от зависти.

Asus C300M 25

Симпатичный хромбук нового поколения, который предлагает мощь и функциональность без наценки за них.

Google Pixel 26

Если у вас набитый карман и слабость к хромбукам, эта модель для вас.

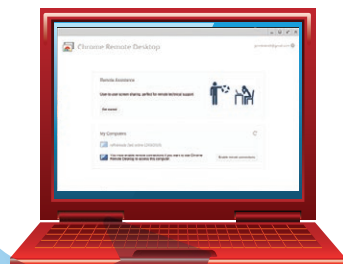
Ubuntu 15.04

Уже с Systemd

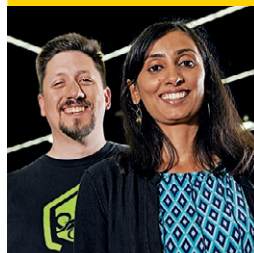
Откроем все новые функции этого знакового релиза с. 36



Сравнение: Клиенты удаленной работы с. 30



Люди говорят



“ Ты не просто пишешь код, а делаешь что-то для общества. ”

Как PayPal заразился открытым кодом с. 44

На вашем бесплатном DVD

Побалуйте себя и любимых подпиской на LXF!

Ubuntu 15.04
СВЕЖИЙ РЕЛИЗ
Новые для 2015!
» Готов к работе
» Готов к установке
» Обновиться сегодня!
32 & 64-битный

ПЛЮС: ANDROID 4.4 X86
ВСЕ ЧТО ТРЕБУЕТСЯ ДЛЯ СТАРТА В LINUX

**Ubuntu 15.04 64-битный,
Ubuntu 15.04 32-битный,
Android 4.4 x86**

» Каждый месяц — только лучшие дистрибутивы

ПЛЮС: Hotpicks, Сравнение и более того! **с. 106**



Доступно в AppStore!



www.linuxformat.ru/subscribe

Для пользователей Raspberry Pi



Новости Pi 88

Приветствие от Лиз Аптон и происшествия в Фонде Pi.

Обзор: HAT Pro 89

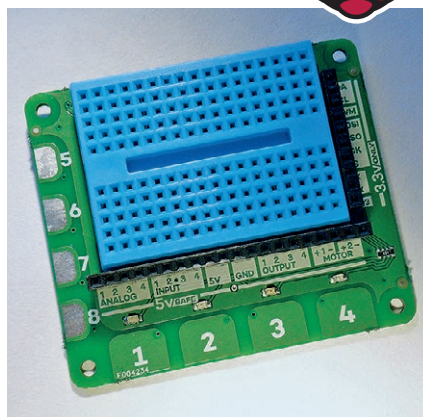
Новое расширение от Pimoroni: стоит ли вам на него раскошелиться?

ЖК-диоды и GPIO 90

Новичок в Raspberry Pi? Хотите научиться программировать GPIO? Вот идеальный проект для старта.

Строим сервер на Pi 2 92

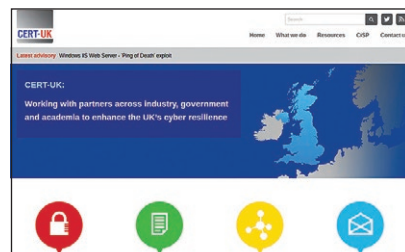
Реализуем потенциал нового Raspberry Pi 2 и запустим полноценный Linux-сервер.



В этом номере...

Защитим Linux! 50

Но Linux же безопасен, верно? Неверно. Разберемся с текущей сворой эксплойтов и зловерного ПО в Linux и посмотрим, как себя обезопасить.



» На ваш сервер наезжают хакеры.

Академия коднга

Аргументы команд 76

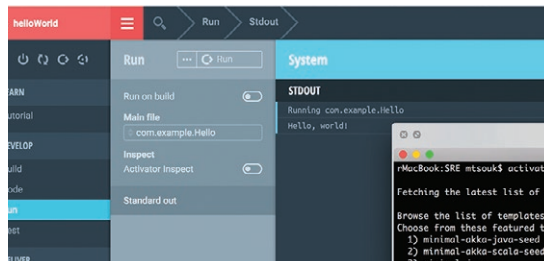
Веские аргументы в пользу более продуктивной работы.

Букварь по Scala 80

Этот гибкий язык функционального программирования способен на многое.

Игрострой с Unity 5 84

Срочно за разработку игр! Благодаря IDE в Linux не придется писать ни строчки кода.



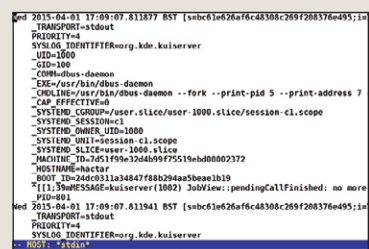
Учебники

Производительность Sysstat 60

Вечный вопрос сисадмина о производительности получает углубленный ответ.

Systemd Почитаем Journalctl 64

Не только наш журнал достоин прочтения: системные журналы тоже полезны.



» Демоны спешат на выручку.

Азы терминала Мошь sudo 66

Добейтесь прав суперпользователя, и власть над системой будет в ваших руках.

Продвинутый мониторинг SystemTap 68

Практическое руководство по системе мониторинга — здесь есть все, что вам надо знать.

Оборудование Взломаем роутер 72

Взломаем, но в хорошем смысле: чтобы наделить его мощью встраиваемой открытой ОС DD-WRT.

Постоянные рубрики

Новости 6

Закрытые ОС — те же вирусы, вещам выдали ОС, Blackberry шерстит персонал, смартфоны шлют фото в Web, разработчики воруют нос от Win10, Google анализирует данные облаков, у ВQ — убунтофон №2, а у РОСА — ОС.

Репортаж: Geek Picnic 12

Александр Толстой попал на пикник для гиков и делится впечатлениями.

Новости Android 28

Android едет на авто, Lollipop помаленьку распространяется, смартфоны запоминаются, а iOS обгоняет Android.

Сравнение 30

Трудиться можно и издалека — была бы работа не виртуальной,

а реальной. *Chrome Remote Desktop, NoMachineX, Remmina, TigerVNC, X2Go.*

Интервью LXF 44

Пурнима Венкатакришнан и Седрик Уильям из PayPal понимают, что работа с деньгами требует открытости. Но не препятствует открытому ПО.

Рубрика сисадмина 54

У опоздания Джолиона Брауна уважительные причины: он завлекал женщин в сисадмины и вскрывал контейнеры.

Ответы 96

ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!
Нейл Ботвик — про настройки Wi-Fi, права доступа к каталогу, установку openSUSE, пакеты разработчиков, монтирование сети, запись действий.

HotPicks 100

Отвдайте горяченького! Лучшие в мире новинки свободного ПО: *Audacity, Brackets, Fiziko, FreeCol, FreeMind, Geary, Linphone, Netsurf, Pasang Emas, Shotwell, VirtualBox.*

Диск Linux Format 106

Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108

Безумцы, вы не читали про Tor и Tails? Скорей бегите покупать, пока есть что!

Через месяц 112

Досконально разберемся в очень важном релизе Ubuntu — 15.04: отныне там поселился *systemd*, да и не только.



В ЭТОМ НОМЕРЕ: Кто у нас вредитель » От телефонов к вещам » ОС для IoT » Фото — в облака » Windows 10 не влечет » Большие данные » 2-й телефон с Ubuntu » Опять ROSA

ЗА СВОБОДУ

Так говорил Ричард

Столлмен объявил Windows и Apple OS X вредоносным ПО.

Самый, пожалуй, известный борец за свободу ПО, Ричард Столлмен [Richard Stallman], следующим образом охарактеризовал некоторые популярные ОС и сервисы: «Windows и Apple OS X суть вредоносное ПО, Amazon — сюжет для Оруэлла, а всякий, доверяющий Интернету вещей — не очень умён (в оригинале Столлмен употребил изящное “an ass”)». В колонке издания *The Grauniad* Столлмен проповедует далекий от техники массам о вреде проприетарного ПО и привязки к производителю, и как кодирование за закрытыми дверями упрощает тайные соглашения шпионских агентств с национальными государствами.

По мнению Столлмена, вредоносное ПО — это в первую очередь ОС. «Системы

Apple — MacOS и iOS — содержат malware. Android также содержит malware в своих несвободных компонентах: бэкдоры для удаленной принудительной инсталляции или удаления любых приложений». Ссылаясь на отчёт Блумберга, Столлмен обвиняет Microsoft в «саботаже» пользователей Windows: информация об уязвимостях передается в АНБ до выпуска исправлений.

Как “malware” Столлмен заклеймил не только Windows и MacOS: куклы Барби, Smart TV и автомобили тоже подверглись его ярости. Ведь потенциально маркетологи могут получить доступ к тайным страхам ребёнка, или к разговорам в гостиной!

Продукт с открытым исходным кодом тоже никак не застрахован от уязвимостей, но присущая такой модели прозрачность

означает, что недостатки, скорее всего, найдут и исправлят. Перспектива вставки бэкдора в код становится рискованным предприятием, поскольку его значительно легче обнаружить. Но и в окружении вредоносных смартфонов, социальных сетей и интернет-устройств оказать сопротивление, по Столлмену, возможно:

- » индивидуально — отказываясь от проприетарного ПО и web-сервисов;
- » коллективно — организовав или поддерживая разработку открытых альтернатив, не отслеживающих пользователя;
- » демократически — законом закрепив уголовную ответственность за вредоносные практики. При этом нужен отказ от соглашений, подобных TRP и TTIP, позволяющих вендорам подавлять демократию.



» Рубрику готовил
**АНДРЕЙ
ГОНДАРЕНКОВ**

ИНТЕРНЕТ ВЕЩЕЙ

“Brillo” от Google

Представлена ОС для встраиваемых устройств.

На конференции для разработчиков Google I/O 2015 (28 и 29 мая) была представлена ОС, ориентированная на компактные встраиваемые устройства с низким энергопотреблением, образующие т.н. «Интернет вещей». Новая свободная ОС от Google, названная “Brillo”, обеспечит для таких устройств обработку процессов начальной загрузки, ввода/вывода и сетевой коммуникации.

Android в его нынешней форме не приспособлен для подобных задач, поскольку, например, версии для смартфонов требуются как минимум 512 МБ ОЗУ. Brillo предназначена для устройств с 64 МБ или

даже 32 МБ памяти, у которых могут отсутствовать собственные дисплеи.

Это не первая и не единственная инициатива Google для умных домов и Интернета вещей. В 2011 г. компания представляла платформу “Android@Home”, в рамках которой Android-аппараты взаимодействовали с домашними приборами и электроникой. Но руководство компании вскоре пересмотрело свои планы по развитию того проекта, посчитав, что «время ещё не пришло, и рынок не готов». Компания Google также владеет производителем «умных» термостатов Nest, продукты которого интегрируются в Умный дом.

Необходимость создания ОС для Интернета вещей несомненна, хотя Google пытается выйти на очень и очень заполненное поле. Над версией Windows 10 для IoT трудится Microsoft. Производителям оборудования, работающим с iOS, предназначен фреймворк Apple HomeKit. Компания Particle предлагает не только ПО, но и аппаратные средства. В конце мая свою легковесную ОС для IoT анонсировала компания Huawei. Но в долгосрочной перспективе, такая конкуренция — благо для конечных пользователей.

» Новая свободная операционная система Google ориентирована на компактные встраиваемые устройства с низким энергопотреблением.



БИЗНЕС

Новая стратегия BlackBerry

Компания перераспределяет ресурсы прибыли ради.

Чтобы повысить прибыльность бизнеса, канадская компания BlackBerry планирует сократить некоторое количество персонала, занятого производством аппаратных средств, одновременно расширив штат других подразделений, в частности, работающих с Интернетом вещей. В компании уверены, что для BlackBerry наступил очередной благоприятный этап, и поэтому необходимо перераспределение ресурсов по тем направлениям, которые лучше всего позволяют извлечь выгоду из возможностей роста, обеспечив устойчивую прибыль во всех областях бизнеса.

BlackBerry планирует расширение в направлении таких сфер, как корпоративные рынки и безопасность. Например, компания запустила кросс-платформенную технологию мобильного управления предприятием BES12 и объявила о партнерской программе с Samsung Electronics по интеграции BES12 со смартфонами Galaxy и планшетами со встроенной технологией безопасности Samsung Knox. Кроме того,



BlackBerry предприняла попытку заполучить долю рынка небольших встраиваемых устройств с беспроводным подключением — представила платформу IoT Platform, первоначально ориентированную на автомобильную и отслеживающую отрасли.

В марте компания объявила о завершении перехода к организационной структуре в виде производственных подразделений (бизнес-устройства, корпоративные сервисы, бизнес-технологии и обмен сообщениями). Недавними приобретениями

► Для повышения эффективности бизнеса канадская компания перешла к новой организационной структуре и перераспределяет персонал среди своих производственных подразделений.

BlackBerry являются немецкая компания Secusmart (провайдер средств шифрования речи и данных, а также технологий антиподслушивания для правительственных организаций, предприятий и провайдеров услуг связи), и английская компания Movirtu, провайдер технологии виртуальной идентификации для операторов мобильной связи.

По состоянию на 28 февраля, конец финансового года, в BlackBerry насчитывалось 6225 штатных сотрудников. За истекший финансовый год компания выпустила 4 новых модели смартфонов BlackBerry 10, включая Classic, Passport, Z3 и Porsche Design P'9983, однако доля BlackBerry OS на рынке снизилась. По данным исследовательской компании IDC, в первом квартале текущего года она составляла всего 0,3%. Доходы BlackBerry в квартале, окончившемся 28 февраля, составили \$660 млн. Из них около 42% приходится на аппаратные средства, 47% — на сервисы и 10% — на программное обеспечение.

ОБЛАЧНЫЕ ТЕХНОЛОГИИ

Разгрузим смартфоны

Google выпустила приложение для сохранения видео и фото.

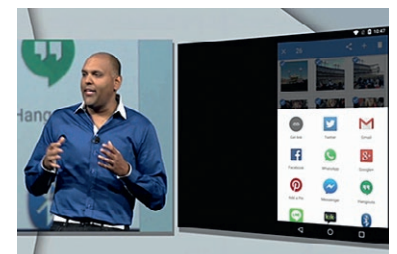
Приложение Google Photos на I/O конференции компании в Сан-Франциско представил ведущий менеджер проектов Google Анил Сабхарвал [Anil Sabharwal]. По его словам, основное назначение Google Photos — упростить пользователям управление огромным массивом медиа-данных, генерируемых с их смартфонов. С помощью Google Photos пользователи смогут загружать в облако изображения с разрешением до 16 Мпикс и видео в качестве 1080p. Поддерживается работа с устройствами на платформах Android и iOS, а также настольные web-браузеры.

Предусмотрена возможность сортировки изображений по дате, персоналиям или месту съемки. Google Photos также использует поисковую технологию компании, выдающую результаты по ключевым словам или ситуационному запросу,

например, фото, сделанные во время «метели в Торонто». Функция Assistant, используя машинное обучение, предлагает новые способы группировки изображений в библиотеке фотографий пользователя. Функция позволяет автоматизировать монтаж изображений и видеоклипов в слайд-шоу и видеофильмы, с возможностями последующего редактирования и совместного использования.

Google упростил совместное использование и пакетную обработку фотографий, создав с этой целью новый жест для сенсорного экрана. Пользователь нажимает и удерживает одно изображение, а затем перетаскивает палец через ряд миниатюр в приложении, выбирая их все сразу. После этого генерируется ссылка, которую можно отправить друзьям и для использования которой не требуется вход в систему.

► Фотографируйте своим смартфоном направо и налево: Google Photos забросит ваши снимки в облако.



Онлайн-компонент хранилища подобен анонсированной в прошлом году библиотеке Apple iCloud Photo, позволяющей пользователям хранить свои фотографии, резервные копии их устройств iOS и другие файлы в облаке Apple. Однако Google предлагает свой безлимитный сервис хранения фотографий бесплатно. Apple же в iCloud предоставляет бесплатно только 5 ГБ, а за дополнительный объем хранилища придется выкладывать деньги.

КОНКУРЕНЦИЯ

Кому он нужен, этот...

Разработчики приложений для Android и iOS не заинтересованы в Windows 10.

Известно, что мобильная ОС Microsoft следующего поколения позволит выполнять приложения для Android и iOS, предоставив инструментарий для портирования существующего кода на Windows 10. Однако здесь главной проблемой для Microsoft является незаинтересованность создателей программ. Агентство Reuters опросило более дюжины разработчиков, создающих приложения для мобильных платформ Apple и Google, и ответ был единодушен: учитывая скромную долю рынка Windows Phone, они не видят необходимости в портировании своих приложений для этой ОС.

Шон Орелли [Sean Orelli], компания Fuzz Productions: «Windows Phone должна будет получить значительную долю рынка, пока не станет чем-то, что экономит нам



► Логическая игра *Candy Crush Saga*, ставшая хитом на платформе Android, будет предустановлена в Windows 10.

время и/или деньги». Джейсон Тэйн [Jason Thane], генеральный директор General UI, в качестве причины, почему его компания не планирует портировать существующие приложения для Windows 10, назвал также стоимость: «Стоимость портирования на новую платформу составляет 50 и более процентов от стоимости оригинальной разработки. Будь у Microsoft способ сделать это легко и дешево, без серьезного снижения производительности или

функциональности, у них не было бы отбоя от желающих».

Разработчиков беспокоит, что новый инструментарий Microsoft будет сложнее в использовании, чем ныне применяемый их компаниями. Эрик Руккер [Erik Rucker], глава мобильного подразделения компании Smartsheet, в беседе с Reuters выразил сомнение, что настройка приложений для iPhone или iPad будет так же проста, и добавил, что в итоге его компания будет вынуждена писать «целую кучу лишнего кода». Но нашлись и разработчики, заинтересованные в Windows 10. Один из них — King.com, создатель красочной головоломки *Candy Crush Saga*, которую предустановят в Windows 10. О создании приложений для Windows 10 объявили также Walt Disney, Netflix и USA Today.

НОВЫЙ ПУТЬ

«Облачная» стратегия Google

Главное внимание компания уделяет анализу «больших данных».

Руководители компании Google заявляют: «Мы не смогли пока догнать Amazon на „облачном“ рынке, но ситуация не вызывает тревоги». В сфере облачных вычислений Google придерживается несколько иного направления: помощь клиентам в анализе их данных. «Это не погоня за Amazon, — заявил в ходе Конференции I/O Брайан Стивенс [Brian Stevens], вице-президент Google по облачной платформе. — Это не просто перемещение серверов в облако. Это не выполнение [в облаке] тех же самых вещей, которые уже выполняются. Это новый облачный путь».

Такая стратегия может изменить способ ведения бизнеса компаниями, а также превратить Google в «облачный локомотив». По мнению Стивенса, облачный сервис Google должен не только избавить предприятия от необходимости приобретать свои собственные серверы, и лучше, чем где-либо, обеспечивать сохранность данных и поддержку оборудования: при использовании облаком компании должны иметь возможность извлекать больше информации из своих данных.

«Всё сводится к анализу данных, — уверяет Стивенс. — Компании необходимо лучше понимать своих пользователей. Чего они хотят? Что им нравится? На основе собранной информации компания сможет построить свою стратегию. Как только приходит понимание, каким образом можно использовать данные в облачных вычислениях, правила игры меняются. Первые же результаты, полученные из анализа данных, всегда впечатляют. И если первоначально целью компании видится присутствие в Web и организация в облаке нескольких серверов, то затем приходит понимание, что возможна интеграция в аналитику больших данных».

Предоставление клиентам возможности разобраться в массивных загрузках данных [big data] — вот что отличает предложение Google. «Благодаря такой стратегии предложения Google отличаются от аналогов Amazon и большинства других облачных провайдеров, — пояснил Дэн Олдс [Dan Olds], аналитик компании Gabriel Consulting Group. — Наличие у Google простых и удобных для пользователя аналитических инструментов будет

► Продуктивную аналитику больших данных Google делает краеугольным камнем своей облачной стратегии.



иметь реальное преимущество перед остальными пакетами».

Стремление Google сфокусироваться на предоставлении своим облачным клиентам возможности анализа больших данных не удивительно: компании необходимо дифференцироваться от AWS, доминирующей на облачном рынке. По мнению Олдса, подход Google заключается в предоставлении инструментария для работы с аппаратными средствами, программным обеспечением и аналитики. Это вовсе не означает, что пользователям AWS недоступен анализ данных. Просто Google делает аналитику краеугольным камнем своей облачной стратегии.

7 советов быстрого поиска работы от hh.ru

- 1. Определите цель.** Решите, кем вы хотите работать, как бы смешно это ни звучало. Точно сформулируйте вашу должность. Работодатель не найдет вас, если название резюме будет общим: «Менеджер» или «Начальник». Лучше уточните: «Менеджер по закупкам» или «Начальник строительной бригады».
- 2. Узнайте о своих способностях.** Вы все еще в поиске своего призвания? Пройдите онлайн-тест «Профориентация»* hh.ru/article/proforientation_promo и узнайте, какая работа вам больше всего подойдет.
- 3. Составьте резюме.** Сделать это на hh.ru легко. Главное — заполните все предлагаемые поля. Уделите особое внимание опыту работы и вашим достижениям — так вы покажете работодателю вашу компетентность.
- 4. Настройте процесс.** Подпишитесь на подходящие вакансии и получайте самые свежие на почту. А также скачайте мобильное приложение HeadHunter, чтобы искать работу в любое время, в любом месте.
- 5. Действуйте.** Откликайтесь на все интересующие вас вакансии. Пишите сопроводительные письма работодателю, поясняя, почему вас интересует эта вакансия.
- 6. Сделайте резюме заметным.** hh.ru/applicant/services Подключите «Яркое резюме»*, чтобы выделить резюме цветом, и «Автообновление»*, чтобы поднимать его в результатах поиска. Работодатели обратят на вас внимание.
- 7. Подготовьтесь к собеседованию.** Поздравляем, вас пригласили! Самое время подготовить ответ на вопрос: «Почему мы должны взять именно вас?». Узнайте максимум информации о компании и подготовьте небольшую речь о том, какой вы классный специалист.

И помните, что работа найдется для каждого!

КОНВЕРГЕНЦИЯ

Frankenphone как Linux-ПК

Компания BQ планирует второй Ubuntu Phone.

Canonical и испанская компания BQ продолжают продвигать Ubuntu на смартфоны как альтернативу Android и iOS. В конце мая BQ представила второй телефон с Ubuntu, и обе компании работают над смартфоном, сравнимым с ПК. Ранее в этом году BQ выпустила модель Aquarius E4.5 Ubuntu Edition, теперь же в Европе по цене £200 (\$220) начинаются онлайн-продажи Aquarius E5 HD.

Aquarius E5 HD во многом превосходит E4.5 (£170): 5-дюймовый HD-экран (вместо 4,5-дюймового) с разрешением 540×960 пикс, интегрированный накопитель увеличен до 16 Гб (с 8 Гб). E5 HD имеет основную камеру 13 Мпикс, против 8 Мпикс у E4.5. Но E5 HD не поддерживает сеть LTE и оснащён скромным 4-ядерным процессором 1,3 ГГц Cortex-A7 от MediaTek. Процессор ориентирован на смартфоны начального

уровня (по цене менее \$100), и новинке будет нелегко конкурировать с аналогичными продуктами на Android. Главным поводом купить E5 HD станет ОС от Canonical. Интерфейс пользователя убунофона составлен из т.н. «областей [scores]» — домашних экранов, связанных с определенной темой (News, Music, Nearby, Today). Есть и традиционное ПО; телефон умеет выполнять нативные и web-приложения.

BQ считает Aquarius E4.5 успешным, но число проданных устройств не называет. Его доля на рынке смартфонов очень мала: аналитика компании Gartner учитывает продажи убунофонов вместе с такими ОС, как Samsung Tizen и Mozilla Firefox OS. После Android, iOS, Windows и BlackBerry, категория «прочие» занимает в I квартале текущего года 0,4% рынка (около 1,3 млн проданных устройств).

ДИСТРИБУТИВЫ

RED X2 от «НТЦ ИТ РОСА»

Вышла новая версия ОС для корпоративного сектора.

В состав дистрибутива ROSA Enterprise Desktop X2 входит прикладное ПО, решающее большинство типовых задач: офисный пакет *LibreOffice*, web-браузеры *Firefox* и *Chromium*, почтовый клиент *Thunderbird* с интегрированным календарем и планировщиком *Lightning*, обмен сообщениями, просмотрщик изображений *Gwenview*, графический редактор *GIMP*, менеджер фотографий *Digikam*, средства редактирования аудио- и видеофайлов, электронные словари *Stardict* и многое другое; проигрыватель *ROSA Media Player* использует набор кодеков *Ffmpeg*.

Графическая подсистема использует стабильные версии X.org и Mesa. В репозиториях доступны проприетарные драйверы для видеокарт nVidia и AMD (ATI), поддерживается гибридная графика (Intel+AMD, Intel+Nvidia).

Дистрибутив основан на ROSA Desktop Fresh KDE, но предоставляет добавочные

гарантии стабильности: обновления, которые не являются исправлениями уязвимостей и критических ошибок, попадают в ROSA Enterprise Desktop X2 только после тестирования и «обкатки» в ROSA Desktop Fresh. Характеристики системы:

- » Поддержка машин с UEFI и Secure Boot.
- » Работа с GPT-дисками.
- » Поддержка современных файловых систем Ext4 (по умолчанию) и Btrfs.
- » Шифрование системных разделов.
- » Корректная интеграция с другими ОС, уже установленными на компьютере.

Рабочий стол KDE версии 4.14.4 обогащён фирменными доработками: единая точка запуска приложений SimpleWelcome и фирменная панель RocketBar; инструмент быстрого доступа к часто используемым файлам StackFolder; приложение TimeFrame, отображающее активность пользователя на ленте времени.

Для разработчиков включены интерпретаторы популярных языков и СУБД. **LXF**

Новости короткой строкой

» Иски бывших сотрудников Mapdriva после сокращения персонала в 2013 г. вызвали банкротство компании. Источник: www.businessinsider.com

» Комитет ООН по правам человека призвал государства не ограничивать доступ к шифрованию и анонимности. Выгоды от запретов несопоставимы с потерями в области прав человека. Источник: arstechnica.com

» Анонс нового выпуска ядра Linux 2.6.32.67 предупреждает о скорой отмене поддержки этой LTS-ветки (обновляется с 2009 г.). Источник: lkml.iu.edu

» Chrome OS 43 поддерживает автозаполнение параметров кредитных карт в Google Wallet, распознает управляющие фразы "OK Google" на новых Chromebook Pixel и автоматически корректирует ввод с обычной клавиатуры. Источник: googlechromereleases.blogspot.ru

» В Cinnamon 2.6 реорганизован интерфейс настройки, добавлены визуальные эффекты, переделано управление питанием, в файл-менеджере *Neto* появился новый интерфейс управления плагинами. Источник: segfault.linuxmint.com

» На конференции Google I/O 2015 показана очередная редакция мобильной платформы Android M Developer Preview; её выход под номером Android 6 планируется в III квартале 2015 г. Источник: android-developers.blogspot.ru

» Управляющий совет сообщества Ubuntu не сумел сместить Джонатана Риддела [Jonathan Riddell] с позиции лидера сообщества Kubuntu. Источник: jridell.org

» Подбирая типовые пароли, червь Moose проникает через Интернет на портативные устройства с Linux (особо уязвимы Actiontec, Hik Vision, Netgear, Synology, TP-Link, ZyXEL и Zhone). Источник: arstechnica.com

» В Fedora 22 — ядро Linux 4.0, *Yum* сменился *DNF*; при входе в систему по умолчанию *Wayland*; рабочие столы Gnome 3.16, KDE Plasma 5, *Xfce* 4.12. Источник: lists.fedoraproject.org

» Microsoft присоединится к сообществу OpenSSH, чтобы улучшить поддержку в Windows, включить SSH в PowerShell и SSH-клиент «из коробки». Источник: blogs.msdn.com

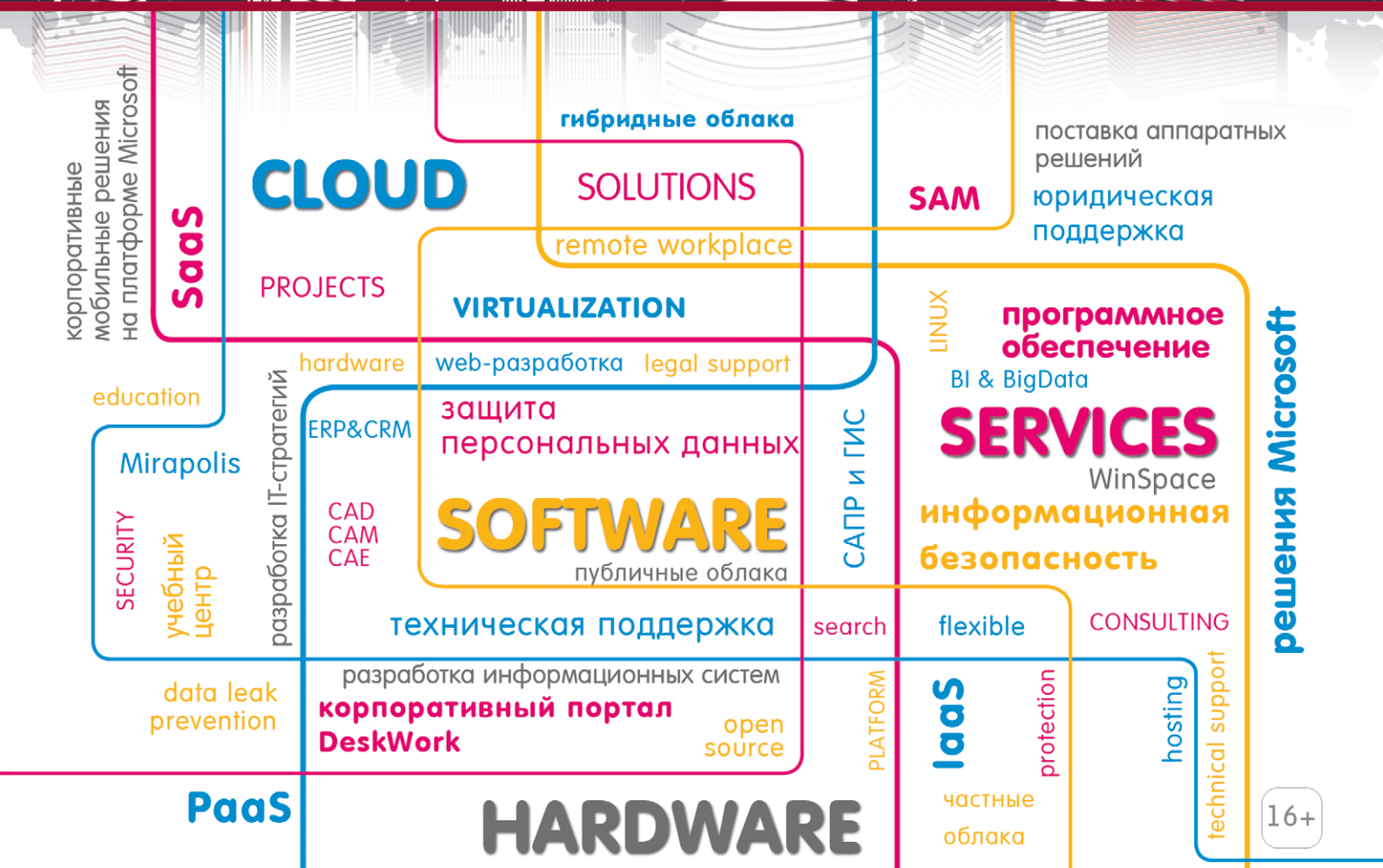
softline®

Cloud Software Hardware Services

20+

Years in IT

IT-архитектура вашего бизнеса





Фестиваль мечтателей

Александр Толстой делится впечатлениями от посещения ставшего традиционным Geek Picnic.

20 и 21 июня на Елагином острове в Петербурге прошёл Гик Пикник — крупное и очень популярное айтишное событие в жизни города и страны. Geek Picnic проходит в Северной столице уже в пятый раз, и за это время небольшой выезд компьютерщиков «на свежий воздух» стал одним из крупнейших open-air лета.

Фестиваль прошёл замечательно и явно удался, но прежде чем погрузиться в детали, впечатления и мысли, стоит кратко рассказать об истории пикника, тем более что ему уже не первый год.

Первое мероприятие прошло в начале августа 2011 г. на лужайке острова Новая Голландия. Целевая аудитория первого пикника — подчеркнутую айтишники, решившие отойти от строгого формата конференций и показать на открытом воздухе устройства, ПО, самодельных роботов и уличные игры. Ключевые темы — образование, карьера, отдых, развлечения. В тот раз зрители, которых собралось не менее 5000, смогли впервые «пощупать» руками настоящий 3D-принтер, покатаются на одноколёсных велосипедах и тут же поиграть в компьютерные игры на приставках.

Через год праздник решили повторить, причём размах стал заметно больше: к прежним развлечениям добавили насыщенную программу мероприятий научно-популярного содержания, лекции

и семинары, удалённые конференции в формате TED, мастер-классы от специалистов российских IT-компаний, неформальную ярмарку вакансий для программистов, открытую библиотеку и много чего ещё. На пикник 2012 г. пришло уже не менее 7000 горожан.

С 2013 г. Geek Picnic проводится ещё более масштабно. Он стал двухдневным мероприятием и переместился на Елагин остров. Вместо прежних двух тематических зон площадку поделили на 4: технологии, наука, искусство, пикник. Участники могли познакомиться с новыми разработками в области робототехники и 3D-принтеров, музеем интерактивной науки, выставкой метеоритов, лабораторией FabLab от питерского Политеха. В каждой из зон, кроме выставок, проходили лекции приглашённых предпринимателей, преподавателей и ученых с мировым именем. Среди них были известный летчик-космонавт Александр Волков, ведущий-преподаватель реалити-шоу «Полиглот» на телеканале «Культура» Дмитрий Петров, инженер-полиглот Анатолий Вассерман, историк Лев Лурье и первый в мире человек с бionicеским протезом руки Найджел Экланд [Nigel Ackland]. Параллельно с лекторием проходил чемпионат России среди квадрокоптеров.

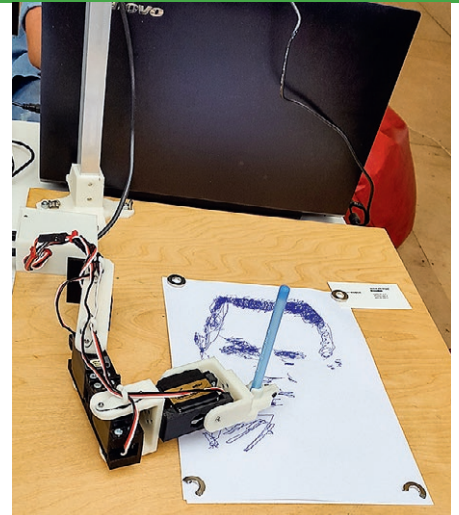
В 2014 г. Geek Picnic прошёл впервые в двух городах: дополнительно к летнему мероприятию

в Санкт-Петербурге присоединилась Москва, принявшая фестиваль зимой в ВВЦ. Всего за два дня на пикнике побывало более 18 000 петербуржцев и более 20 000 москвичей. Спонсором выступила компания PayPal, от которой многим участникам достались фирменные флешки в виде браслетов.

Мы плавно подходим к оценке недавнего пикника, который прошёл 13 и 14 июня в Москве в парке на Красной Пресне и 20 и 21 июня в ЦПКИО им. Кирова в Петербурге. В первую очередь, хочется отметить заметно возросшее число посетителей. Официальных цифр организаторы пока не обнародовали, но по предварительным оценкам



➤ Линуксцентр организовал мастер-класс по робототехнике для всех желающих.



➤ Печать изображений можно выполнять и рукой робота-художника.

самых участников, Geek Picnic в Петербурге в этом году посетило не менее 30 тысяч человек. Публика заполнила весь Елагин остров; это были не только суровые компьютерщики, но и мамы с колясками, иностранные туристы и просто зеваки. Перед входом на Масляный луг, где происходил контроль допуска, вытянулась очередь на 200 м из людей с билетами в руках (абонемент на два дня стоил 750 р.), отдельно была организована палатка для СМИ и людей с персональными приглашениями. Первая сотня метров аллеи внутри Geek Picnic была похожа на оживлённую ярмарочную улицу где-нибудь в Европе. Бесчисленные лавки и передвижные магазинчики предлагали кофе, печенье, коктейли, сладости и закуски, почти всё — авторское, молодёжное, нишевое, ультрасовременное и, как правило, с упором на здоровый образ жизни. Вообще, в этом и прошлом году непропорционально велико было присутствие авторского фаст-фуда, отвоёвавшего заметную часть территории. Рядом с главной сценой, на которой происходили перформансы, лекции и конкурсы, активно жарили мясо на вертеле, пазью, хипстерскую шаверму, вок, предлагали пиво, соки и молочные коктейли. Тут же, в дыму от соседнего шашлыка, активно торговала веганская лавка.

Центральным объектом пикника была огромная робо-рука The Hand of Man от американского художника Кристиана Ристов [Christian Ristow]. Вокруг неё лежали несколько помоях легковых автомобилей (видимо, с автокладбища), которые рука поднимала своими мощными пальцами. 6-тонная конечность, собранная из металлолома,

управлялась человеком с бионическим протезом настоящей руки. Это был учёный и обладатель своей робо-руки Николя Юше [Nicolas Huchet], уже не первый год посещающий Geek Picnic и щедро здоровающийся за руку со всеми желающими!

На центральной сцене прозвучало много выступлений по бионике, киборгизации человечества, нейробиологии. О своём видении человека будущего рассказала учёный-трансгуманист, медиа-художник Наташа Вита-Мор [Natasha Vita-More]. Рассказ певицы и модели Виктории Модесты [Viktoria Modesta] поразил как свидетельство редкой отваги, с которой человек сам выстраивает свою жизнь, не позволив серьёзному увечью наложить на неё отпечаток. С лекциями выступили нейробиолог, нейронинженер и один из создателей теории эмуляции мозга Рэндалл Кёне [Randall

«Центральным объектом пикника была огромная робо-рука.»

А. Коеле) и профессор Университета Калифорнии, философ Крис Хэблс-Грей [Chris Hables Gray], автор «Руководства киборга [Cyborg Handbook]».

Отдельные шатры были у университета ИТМО и Политеха, выставивших традиционные для пикника объекты — всяческие роботы (от мойщиков окон до управляемых аквариумных рыбок), шлемы виртуальной реальности (типа Oculus Rift) и технологии печати. Каждый год на пикнике там и сям встречаются 3D-принтеры, идеально подходящие для изготовления сувениров, а также альтернативные технологии печати: печать обычным, но роботизированным фломастером на бумаге, а то и печать изображения на настоящем блине.

Доступный способ освоить робототехнику предоставил Линуксцентр, организовавший мастер-классы по ScratchDuino для всех желающих. ScratchDuino — это набор-конструктор из датчиков, смонтированных на маленьком колёсном шасси и управляемых с компьютера из среды Scratch. Интерфейс программы специально сделан так, что даже неподготовленные пользователи, в том числе и дети, могут понять базовые принципы программирования и испытать себя в роли «повелителей роботов». По итогам мастер-

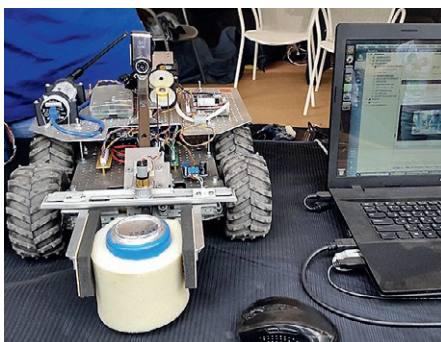
классов победителям вручали памятные призы и подарки, включая свежие номера журнала *Linux Format*.

Рядом располагались уже ставшие привычными развлечения: настольные игры, кинотеатр на пуфиках, верёвочная лестница, фотопечать, люди на джамперах, квадрокоптеры над головами и много-много ролевиков и косплейщиков: от имперских солдат до едва одетых демонических дев.

Эта тенденция сопровождает Geek Picnic уже второй год, как раз с момента взрывообразного роста его посетителей: всё большее внимание обращают на себя второстепенные темы, не относящиеся напрямую к науке и технике. Стало много сувениров, всевозможных «уникальных» товаров, фокусов и типичных уличных развлечений, в то время как сама робототехника и компьютерная индустрия стали немного теряться на заднем плане. Но ничего зазорного в этом нет: богатый выбор развлечений и лекций позволяет точнее ответить на вопрос, кто же такой гик? Итак,

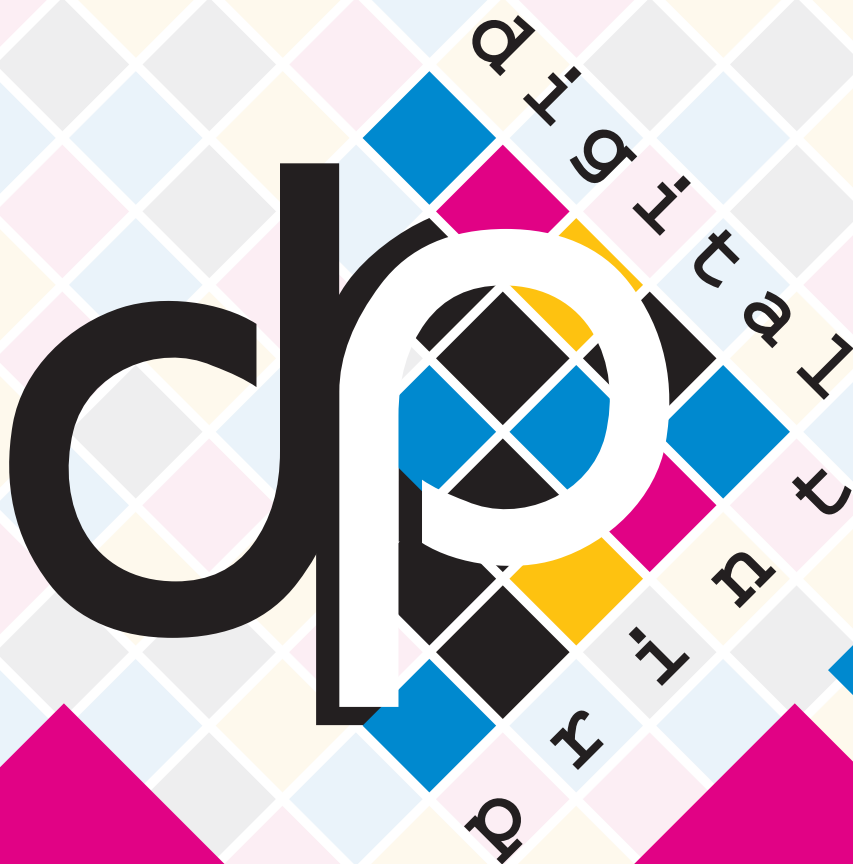
гик — это увлечённый современными технологиями молодой человек или девушка, компьютерщик, поклонник носимой электроники, «Интернета вещей», облачных технологий, настольных игр и ограниченного набора американских сериалов, модно одетый эстет, технофетишист и зачастую интроверт. Нечасто попадающиеся среди этой публики молодые девушки почти всегда стремятся к одному и тому же милому образу — Гаечки [Gadget Hackwrench]: желанной спутницы, в отношении к которой светлое чувство не допускает ни капли пошлости. Однако этот привлекательный и экстравагантный образ гика может показаться отчасти наивным, чему есть свои причины. Обилие недорогих пластиковых протезов, напечатанных на ставших доступными 3D-принтерах, едва ли даёт надежду на реальный транс-гуманизм, а успехи в нейробиологии всё ещё скромны. Не случайно на Geek Picnic-2015 отсутствовал павильон крионики, а Крис Хэблс-Грей заявил со сцены, что скептически относится к возможности досконально изучить мозг бессмертия ради...

Тем не менее, почему бы гикам не жить мечтами и не пользоваться новейшими изобретениями? А фестиваль-2016 состоится в Иерусалиме! LXF



➤ Под управлением компьютера роботоплатформа совершала различные манипуляции.

Оперативная полиграфия



Ризорграфия Lux (600 dpi)

- ◆ Тиражирование на цифровых дубликаторах
- ◆ Инструкции, методички, паспорта продукции
- ◆ Бланки, рекламные листовки, брошюры, книги
- ◆ Самокопирующиеся бланки и проч.

Цифровая печать

- ◆ Визитки, грамоты, дипломы
- ◆ Приглашения, сертификаты
- ◆ Печать переменных данных
- ◆ Буклеты, календари, брошюры
- ◆ Каталоги, листовки, наклейки
- ◆ Послепечатная обработка

Наши преимущества

- ◆ Собственное производство
- ◆ Высокое качество
- ◆ Доставка
- ◆ Минимальные сроки и экономия ваших сил
- ◆ РИА и РА – скидки на всё!

Графический дизайн

- ◆ Полиграфической продукции
- ◆ Наружной рекламы
- ◆ Фирменный стиль
- ◆ Цифровая фотосъемка

Digital PRINT

(812) 591-79-80

(812) 591-79-81

diprint.spb.ru

Санкт-Петербург,
ул. Смольячкова, 19,
офис 405А
(м. «Выборгская»)

LINUX FORMAT

Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов



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

Cinnamon 2.6: триумф non-stop

Выходе версии среды Cinnamon за номером 2.6 объявили 2 июня 2015 г.

В тот же день составляющие её пакеты стали доступны в репозиториях дистрибутивов проекта Mint — будущей Linux Mint 17.2 Rafaela и текущей Linux Mint Debian Edition 2 Betsy. Оно и понятно: разработка Cinnamon почти целиком сосредоточена в рамках этого проекта. А вот почти мгновенная поддержка этой среды майнтейнерами ряда дистрибутивов «первого эшелона» оказалась неожиданностью, как и качество реализации этой поддержки. Прежние версии Cinnamon в «неродных» системах внедрялись неторопливо и подчас не лучшим образом. А сейчас за неделю версия 2.6 появилась в официальных репозиториях Fedora всех поддерживаемых версий (с 20-й по 22-ю, не говоря уже о Rawhide), в репозитории сообщества Arch Linux, в Syslinux'e Alt'a, в semi-official'e openSUSE (как для стабильной версии, 13.2, так и для Thumbleweed)... даже в консервативном Debian'e, хотя и ветке Unstable.

Как ни странно, из этих стройных рядов выбивается Ubuntu — некогда единственный дистрибутив с полноценной поддержкой Cinnamon (кроме родительских, конечно). Ныне же версия 2.6 доступна только в виде «ночных сборок» в PPA-репозитории. Однако и это не в силах омрачить триумфального шествия данной среды. И хотя она пока не удостоилась личной строки в списке «дистрибутирующих» пакетов Distrowatch'a, участи «desktopa одного дистрибутива», каковым является Unity, она успешно избежала.

alv@posix.ru

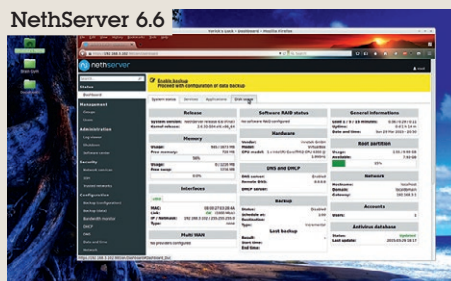
Сегодня мы рассматриваем:

- Bluestar Linux** 16 Возобновляемость релизов — удобство, чреватое проблемами. Пресек ли проблемы данный дистрибутив?
- Blend4Web** 20 Разработчики нашли, чем удивить требовательного пользователя: добавили к Blender экспорт в WebGL.
- SuperX 3.0** 17 Дистрибутив, почему-то скрывающий свое происхождение от открытого кода, и притом с EULA, настораживает.
- Shadowrun Chronicles: Boston Lockdown** 22 Игра в принципе неплохая, кабы не тормозил центральный сервер...
- NethServer 6.6** 18 Слово «серверный» в применении к дистрибутиву больше не должно никого пугать! Разработчики облегчили вам труды по настройке.
- Dell XPS 13 Dev Ed** 24 Смело можно утверждать, что этот ноутбук с Linux на борту способен потягаться на равных с признанными фаворитами.
- Unity 5** 19 Нет, это не тот Unity, про который вы подумали, а игровой движок, для любителей бесплатный.
- Asus C300M** 25 Яркий хромбук с поразительно живучими батареями составит вам хорошую компанию в командировке.



➤ **Asus C300M пленяет красотой и яркостью цветов.**

Chromebook Pixel 26 В Google создали самый мощный хромбук на планете, сумев отместить былые проблемы с батареей. Но и деньги за него придется выложить немалые.



➤ **Кто сказал, что у серверных дистрибутивов враждебная аура? Сумрак переходит в ясное небо.**



➤ **Внезапно пробудившись, главный герой оказывается в совершенно незнакомой обстановке.**

Сравнение: Удаленные рабочие столы с. 30

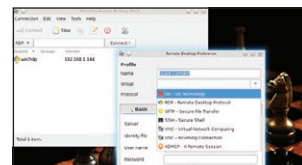
Chrome Remote Desktop



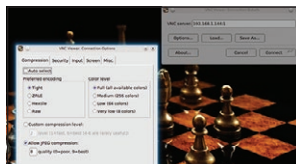
NoMachineX



Remmina



TigerVNC



X2Go



Сидеть летом в душном офисе — не самая приятная перспектива. Но современные технологии позволяют вам рулить своим рабочим ПК хоть бы и с пляжа! Если, конечно, на пляже предусмотрен Wi-Fi.

Bluestar Linux

В этом основанном на Arch дистрибутиве KDE полным-полно приложений. Шашанк Шарма проникает в формулу успеха Deskpro Edition.

Вкратце

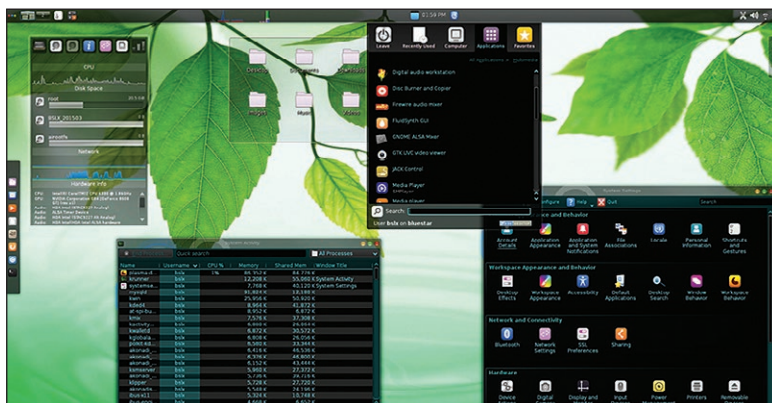
» Дистрибутив с возобновляемым релизом на базе Arch Linux. Имеется в нескольких версиях для архитектуры x86 и x86_64, а благодаря готовым приложениям, куда дружелюбнее к новичкам, чем большинство аналогичных дистрибутивов. См. также: Arch Linux, PCLinuxOS.

Дистрибутивы с возобновляемыми релизами внушают в равной мере восхищение и страх. Немалая часть сообщества пользователей Linux, особенно новички, полагает, что эти дистрибутивы — чисто для умелых рук. Сами дистрибутивы мало что делают, чтобы их в этом разубедить. В отличие от популярной модели возобновляемых дистрибутивов, где надо устанавливать и настраивать каждое приложение самим, Bluestar Linux, с его огромным выбором приложений по умолчанию, идеально подходит как опытным, так и начинающим пользователям.

Основанный на Arch Linux, Bluestar поставляется как устанавливаемый live ISO-образ. Все его редакции, которые почти идентичны, за исключением выбора пакетов по умолчанию, доступны для 32- и 64-битных машин.

Среда live вводит вас в рабочий стол KDE 4.13.3. В отличие от большинства других дистрибутивов KDE, программа запуска Bluestar, *Kickoff*, размещается прямо посреди панели задач, а не традиционно в левом углу.

В Bluestar множество кодеков и плагинов, чтобы вы наслаждались разнообразнейшими мультимедиа-форматами. YouTube и другие сайты потокового видео безупречно работают в имеющихся браузерах *Firefox* и *Chromium*. Кроме того, сюда входит весьма обширная подборка приложений, таких, как *LibreOffice*, *GIMP*, *GnuCash*, плюс ряд интернет-приложений и игр, способных порадовать любую категорию пользователей. Все приложения



» Bluestar Linux Deskpro Edition — отличный выбор для жаждущих урвать дистрибутив с возобновляемым релизом и множеством готовых приложений по умолчанию.

аккуратно распределены по категориям в классической программе запуска *Kickoff*. Для тех, кто не хочет прочесывать меню в поисках любимых приложений, дистрибутив предлагает также мощную программу запуска *Krunner*.

Несмотря на обилие приложений — некоторые из них требуются для дистрибутивов с возобновляемым релизом, например, *Sweeper* и *BitCleaner* для очистки кэша и временных файлов — дистрибутив предлагает гармоничное взаимодействие с рабочим столом. Bluestar впечатляюще отзывчив и неплохо работал даже на старой машине всего с 2 ГБ ОЗУ.

Установи и забудь

Красота дистрибутива с возобновляемым релизом в том, что не надо волноваться по поводу переустановки системы при каждом важном обновлении. Дистрибутив позволяет использовать мощные репозитории Arch Linux, и вы можете обновлять свою систему и приложения без особых усилий.

Установщик Bluestar выглядит примитивным, со своим интерфейсом на базе *Ncurses*. Хотя процесс не очень трудный или требующий вмешательства пользователя, как в Arch или Gentoo, вам придется пройти через этот процесс вручную. Отсутствие документации его усложняет, особенно для новичков. Впрочем, вы можете зарегистрироваться на форуме, где есть информация по разным аспектам дистрибутива, включая установку, обновление и прочие прелести.

Мощный инструмент управления пакетами *Pacman* — это все, что вам нужно

для обновления Bluestar, но дистрибутив поставляется также с надежным интерфейсом *PacmanXG*. Поскольку инструмент пока что в бета-версии, рекомендуем установить какой-нибудь другой популярный интерфейс для *Pacman*, если вы предпочитаете работать с GUI. И снова, отсутствие документации может затруднить начинающих; но, к счастью, данный дистрибутив основан на Arch, а это один из лучших всего документированных дистрибутивов.

Напрашивается сравнение с дистрибутивом PCLinuxOS, поскольку это тоже возобновляемый релиз на базе KDE, но эти двое совершенно непохожи из-за разного наследия и разных систем, лежащих в их основе. Строго говоря, Bluestar тому проигрывает, но только в установке и документации. Не без опаски, но мы его рекомендуем — потому что BlueStar Linux сделан хорошо. LXF

Свойства новскидку

Частые обновления

Почти каждый релиз, помимо обновленной базы и приложений, предлагает обновленный вид и работу.

Текстовый установщик

Установщик Bluestar предлагает только инструменты командной строки, что отпугивает новичков.

LINUX FORMAT Вердикт

Bluestar Linux Deskpro Ed

Разработчик: Джефф Ходд [Jeff Hodd]
 Сайт: <http://bluestarlinux.org>
 Лицензия: GPL и др.

Функциональность	8/10
Производительность	9/10
Удобство в работе	9/10
Документация	6/10

» Bluestar Linux предлагает скорость и эффективность Arch Linux, минус морока с ручной настройкой.

Рейтинг 8/10

SuperX 3.0

Шашанк Шарма сроду не ощущал недостатка в превосходных степенях сравнения, но на SuperX, очередной KDE-дистрибутив, слов ему не хватило.

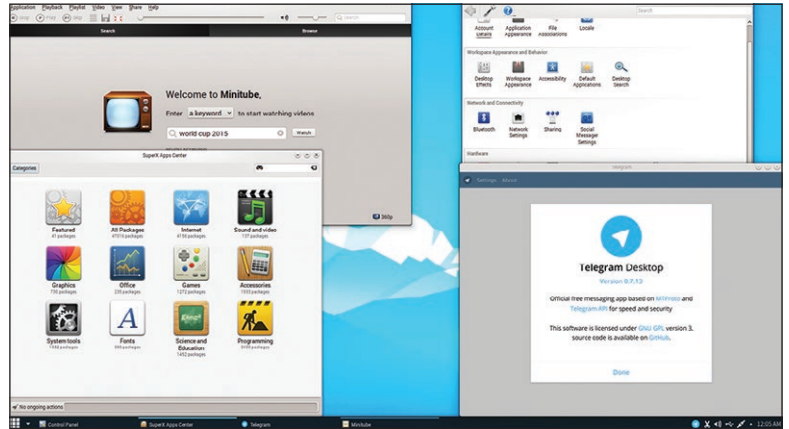
Вкратце

» Аккуратно упакованный дистрибутив с большим выбором мультимедиа- и интернет-приложений. Основан на Ubuntu и KDE, с особым упором на производительность и эстетику. См. также: openSUSE и Manjaro.

Прошедшие несколько лет подарили нам несколько KDE-дистрибутивов, нацеленных и на новых, и на продвинутых пользователей. Некоторые особенно явно направлены на новых пользователей, при множестве примочек, и SuperX — ещё один дружелюбный к пользователям дистрибутив. Он поставляется как устанавливаемый ISO объемом менее 2 ГБ, для 32- и 64-битных машин. В начале установки SuperX предлагает вам Условия обслуживания (VO) [Terms of Service (ToS)], которые вы должны принять, чтобы продолжить установку.

VO недвусмысленно сообщают, что они применимы только к ПО, продуктам, произведениям искусства и т. д., разработанным Libresoft, компанией-производителем SuperX, и не влияют на права, предоставленные пользователям лицензией включённого ПО. Однако мы заметили, что в VO не упоминается о свободном ПО и ПО с открытым кодом или о чём бы то ни было ещё, что хоть как-то намекает на происхождение SuperX от открытого кода, и это серьёзное упущение.

Для дистрибутива, который претендует на улучшенную производительность, у SuperX одна из самых медленных установок из нами виденных. К счастью, процесс разбиения диска на разделы отлично продуман и вряд ли смутит пользователей. У вас есть выбор: выделить весь диск целиком под дистрибутив с или без шифрованного LVM либо назначить разделы вручную. На двудядерной тестовой машине с 4 ГБ ОЗУ вся установка продлилась более 30 минут. Комментарии на сайте также сообщают о периодических сбоях во время



» Несмотря на своё название, здесь нет ошеломляющего X-фактора, достойного рекомендации в качестве дружелюбного к новичкам дистрибутива.

установки с определёнными языками, например, итальянским. Зато после установки SuperX значительно опередил на нашей тестовой машине другие KDE-дистрибутивы, и его впечатляющая коллекция приложений достаточно для большинства пользователей. Хотя инструментам Qt и KDE отдан приоритет и они хорошо интегрированы, включённые приложения GTK тоже не выглядят бедными родственниками.

Готов к бою

Особенностью дистрибутива является двойник программы запуска *Homerun* под названием *SuperX App Launcher*. Ваши любимые приложения и недавно открытые файлы перечислены под вкладкой Home; также вы можете получить доступ к разделённому списку категорий всех установленных приложений из вкладки Application.

С ходу оборудованный для обработки различных типов мультимедиа, последний релиз характеризуется почти годовалым KDE 4.13 с ядром Linux 3.13 и стандартным набором для офисной работы, мультимедиа-приложений и игр.

Дистрибутив претендует на высокую отзывчивость благодаря оптимизации управления памятью, даже при малом ОЗУ, и меньшей зависимости от подкачки. Ключевая особенность релиза — предзагрузка часто используемых приложений, что в результате даёт их более быстрый запуск. К сожалению, в этом отношении дистрибутив оказался не быстрее, чем любой другой современный дистрибутив.

У нас не случилось никаких сбоев, и мы не обнаружили замедления отклика даже

при нескольких запущенных приложениях. Форум очень активный, но вики достаточно скудная и содержит устаревшую информацию, так что требует вмешательства от разработчиков и сообщества.

Явных проблем с этим дистрибутивом нет, но мы с подозрением относимся к любому проекту, скрывающему свою свободу и происхождение от открытого кода и заставляющего пользователей принимать лицензионное соглашение [EULA]. Этот релиз вышел почти через два года после SuperX 2.0, но сайт не предлагает никакой информации о модели разработки дистрибутива. Хуже того, первоначальные релизы были основаны на Ubuntu, а сайт этого факта не признаёт. Для заинтересованных есть куда более прозрачные и динамичные дистрибутивы KDE, такие как openSUSE, OpenMandriva и Manjaro Linux. **LXF**

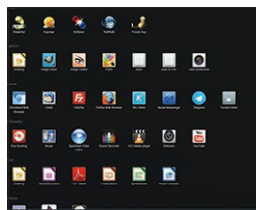


Свойства навскидку



Центр развлечений

Дистрибутив имеет широкий выбор мультимедиа-приложений, которые помогут вам развлечься.



Производная Ubuntu

Дистрибутив основан на Ubuntu, и вам можно использовать лежащую в основе систему `dpkg` и `apt-get`.

LINUX FORMAT Вердикт

SuperX 3.0

Разработчик: Libresoft Technology Pvt Ltd
Сайт: <http://superxos.com>
Лицензия: GPL и др.

Функциональность	7/10
Производительность	8/10
Удобство в работе	8/10
Документация	5/10

» SuperX не предлагает ничего особенного для новичков и не даёт основания перейти на него с других KDE-дистрибутивов.

Рейтинг 6/10

NethServer 6.6

За последние годы запуск серверов упростился. Но **Шашанку Шарме** нравятся дистрибутивы типа NethServer, которые эту задачу облегчают дополнительно.

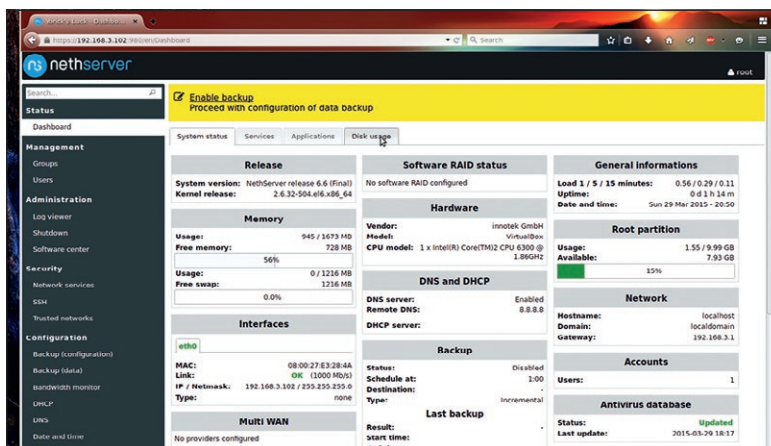
Вкратце

» Спецдистрибутив на основе CentOS для серверов. Обычные серверные компоненты, такие как web, почта, файловый сервер и т.д., доступны в виде модулей, пригодных для расширения дистрибутива. NethServer также предлагает web-интерфейс для обычных задач администратора.

По сравнению с настольными дистрибутивами, которые извечно трудятся над тем, чтобы сделать установку и настройку как можно дружелюбнее к новичкам, серверные дистрибутивы излучают довольно враждебную ауру. Однако дистрибутивы вроде NethServer прилагают огромные усилия, чтобы это изменить. NethServer — серверный дистрибутив на базе CentOS, который помогает настроить любой тип сервера — web, почтовый, FTP и т.д. — без возни со сложными файлами настройки. Практически все аспекты своего сервера можно настроить через web-интерфейс. Дистрибутив доступен только для машин x86_64 в виде устанавливаемого ISO-образа, поболее 600 МБ, и имеет три режима установки — интерактивный, автоматический и стандартный CentOS. Первая опция, несмотря на свое название, не предоставляет выбора устанавливаемых пакетов, но позволяет указать язык установки, настройки RAID и пароль администратора системы и настроить сеть. Автоматический режим устанавливает NethServer с настройками по умолчанию. Ни одна опция не даст назначить файловую систему для раздела root или схему деления на разделы.

Менеджер сервера

После установки можно войти в систему с паролем по умолчанию "Nethesis,1234" как root. Затем вы сможете получить доступ к сетевому интерфейсу NethServer, который именуется Server Manager, с любого компьютера в вашей сети, направив



» NethServer — полнофункциональный и стабильный серверный дистрибутив.

браузер на <https://192.168.3.120:980> (IP-адрес подставьте для своей установки NethServer). Для входа в Server Manager воспользуйтесь именем и паролем пользователя root системы. В интерфейсе Server Manager все уведомления, например, о текущих задачах, вроде запланированного резервного копирования, установки программ, или подробности об имеющихся обновлениях, отображаются ярко-желтым цветом.

Установка по умолчанию предоставляет только базовую систему. Все серверы, с которыми вы хотите работать, придется установить вручную. Но благодаря Software Center дистрибутива обычно сложный процесс установки, скажем, почтового сервера сводится всего к паре щелчков.

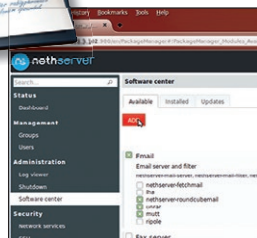
Software Center, расположенный под заголовком Administration на боковой панели, предоставляет вам список установленных и доступных пакетов. NethServer рассматривает разные серверы, web-, почтовый и т.д., как модули. При установке модуля дистрибутив автоматически скачивает все необходимые зависимости. К сожалению, вам не сообщают об общем числе зависимостей заранее, и скачивание/установку после того, как она началась, нельзя отменить. Модули предоставляются для всех популярных компонентов сервера: почта, web, файлы, сервер IM, брандмауэр, прокси-сервер и т.д. Парой щелчков можно настроить ownCloud или VPN.

Имеется также масса полезных инструментов, которые помогают анализировать свой сервер и его компоненты. Log Viewer перечисляет все имеющиеся лог-

файлы, которые для каждого вновь установленного сервиса добавляются в список автоматически. Строка поиска позволяет найти совпадения в лог-файлах практически мгновенно. Еще одна полезная функция — Disk Analyser, она предоставляет график использования диска каждым контроллером. Вы можете использовать этот дистрибутив как фильтр контента для блокировки определенных расширений файлов, фильтрации URL по совпадению вырежений и многого другого.

Другие серверные дистрибутивы, например, Zentyal, тоже предлагают web-интерфейс, но NethServer выделяется из них благодаря скорости и простоте. Его интерфейс прост в навигации и не перебарщивает с числом опций настройки. Добавочный бонус — динамичное сообщество пользователей, готовых вам помочь. **LXF**

Свойства новскидку



Web-интерфейс

Боковая панель расширяется по мере добавления модулей, чтобы предоставить быстрые ссылки на страницы настройки.

Info	Application	L4 Proto	Client
SSDP	UDP	192.168.0.10294	
Unknow	UDP	192.168.0.10291	
Unknow	UDP	192.168.0.11017000	
Unknow	UDP	192.168.0.11017000	
HTTP	TCP	192.168.0.11044075 (\$ 148 ms)	
HTTP	TCP	192.168.0.11044075 (\$ 148 ms)	
HTTP	TCP	192.168.0.11044077 (\$ 167 ms)	
HTTP	TCP	192.168.0.11044080 (\$ 172 ms)	
HTTP	TCP	192.168.0.11044081 (\$ 172 ms)	
Unknow	UDP	192.168.0.11041000	
SSDP	UDP	192.168.0.11044716	

Богатый ассортимент

Дистрибутив может похвастаться множеством полезных инструментов, таких, как Log Viewer, Disk Usage Analyser, и т.д.

LINUX FORMAT Вердикт

NethServer 6.6

Разработчик: Nethesis SRL
 Сайт: www.nethserver.org
 Лицензия: GPL

Функциональность	9/10
Производительность	9/10
Удобство в работе	9/10
Документация	8/10

» Полноценный серверный дистрибутив, разработанный для того, чтобы сделать настройку самой простой частью создания сервера Linux.

Рейтинг **9/10**

Unity 5

После долгих лет ожидания освещения, достойного AAA-игры, Робин де Йонг ошеломлен чудесным преобразованием уровней своей корявой поделки.

Вкратце

- » Кросс-платформенный
- » AAA-графика
- » 3D-редактор
- » 2D-редактор
- » Инструмент анимации Mecanim
- » Поддержка C# и JavaScript
- » Бесплатная и без выплат авторам
- » Для групп разработчиков требуется оплата лицензии
- » Отличное сообщество и документация

Небогатым разработчикам инди-игр сроду так не везло. На конференции разработчиков игр *Unreal Engine 4* сделали бесплатным, Valve представила долгожданный движок Source 2 и вышел Unity 5. Unity — движок для создания игр, позволяющий разработчикам и художникам создавать видеоигры с GUI, похожий на тот, что вы используете в пакете трёхмерного моделирования, и объединять его с C# или JavaScript для определения сценария поведения объектов в игре. Созданную игру можно экспортировать как исполняемый файл, включающий движок, необходимый для запуска графики, звука, действия пользователя, анимации, физики, и т.д. У Unity 5 есть куча новых фишек, впервые предоставленных пользователям бесплатного Personal Edition. Главным нареканием на Unity 4 была недоступность функции мягких теней без применения платных обновлений. Для инди-разработчиков это значило, что надо изыскивать обходные пути, лишаясь разумного и эффективного по времени рабочего процесса. Хуже того, чтобы извлечь максимум пользы из графики, разработчику приходилось учиться кодировать на высокоуровневом языке шейдеров — High Level Shader Language (HLSL).

Войдите в Unity 5 в Global Illumination (GI, глобальное освещение) в реальном времени и в новый Standard Shader. GI берёт световой эффект, предусмотренный в программе рендеринга, и применяет его к игровой среде в реальном времени. Он высчитывает вторичный блик, который можно автоматически привязывать к текстурам игры или высчитывать динамически во время запуска. Со Standard Shader

» С основанными на физике шейдерами, ощутима прибавка реализма — без обращения к коду.



» Непрямые блики позволяют настроить освещение и предоставить Unity сделать всё остальное.

можно задавать материалы, чтобы имитировать практически любой реальный материал, не прикасаясь к коду. Эти два достижения открывают новые рынки — в голову приходит Archviz — и снижают барьеры для инди, позволяя им внести в свои игры реализм AAA-уровня.

Поддержка платформы

Причина распространенности Unity в том, что он портирует один и тот же код в разные платформы. Возможен экспорт в Linux, iOS, Android, BlackBerry, PC, OS X, Windows Phone 8, Web player, WebGL, Xbox 360, Xbox One, PS3, PS4, PS Vita и Samsung TV; а как насчёт движения в обратном направлении в канале контента? Unity 5 импортирует форматы файлов .FBX, .dae (Collada), .3DS, .dxf и .obj, а также проприетарные.

Mecanim, система конструирования анимации Unity, допускает анимацию объектов через клипы из внешних источников, созданных сторонними инструментами или приобретенных у съемочных студий. Увы, в Unity нельзя создать собственную анимацию. Физика работает по умолчанию с настройками для сталкивающихся объектов, твёрдых тел, суставов и тканей, и приятно, что для 2D-игр есть отдельный движок физики. Имеются системы частиц, создатель деревьев, ландшафтные инструменты, редактор спрайтов и инструменты UI (или GUI). Придираться трудно, особенно учитывая бесплатность всего этого. Но в чем же недостатки? Хотя Unity экспортирует игры для работы на Linux, SDK работает только на Windows или Mac. Некоторым удалось запустить её

в Wine, но эта настройка — сущее мучение. Нельзя похвастаться обилием документации или мощной поддержкой сообщества. Мы можем сравнить её с Epic, который начал портировать UE4 в Linux и весьма серьёзно отнёсся к поддержке его пользовательской базы (см. https://wiki.unrealengine.com/Linux_Support). Кроме того, у GI и Standard Shader не хватает ряда простых преднастроек, которые значительно облегчили бы навигацию по всему массиву настроек. Пользователей, которые боятся скриптов, расстроит отсутствие инструментов визуального создания скриптов, таких как Blueprints от Unreal Engine и Flowgraph от CryEngine. Правда, онлайн-документация хороша. Если вы согласны мириться с этими ограничениями, хорошая новость в том, что Unity отказалась от модели уплаты авторских отчислений. LXF

LINUX Вердикт
FORMAT

Unity 5

Разработчик: Unity Technologies
Сайт: <http://unity3d.com/5>
Цена: Бесплатно, для профессионалов \$75/месяц

Функциональность	7/10
Производительность	7/10
Удобство в работе	6/10
Документация	4/10

» Unity 5 усиливает реализм благодаря глобальному освещению, подготовкой текстур в реальном времени и основанным на физике шейдерам.

Рейтинг 9/10



Blend4Web

WebGL наступает! **Андрей Прахов** с удивлением узнал о возможности создания HTML5 силами *Blender*.

Вкратце

» Blend4Web — интерактивный WebGL-движок от отечественных разработчиков. Быстрый, маленький, хорошо документированный. См. также: Unity3D, PlayCanvas, GLGE.

Технология WebGL стремительно наступает. Её активно поддерживают современные браузеры. Стало модным совмещать интерактивные элементы с дизайном сайта, а просмотр трехмерных моделей в каталогах интернет-магазинов уже обыденная и удобная возможность.

Спрос рождает предложение. Неудивительно, что появляются все новые и новые программные решения для создания WebGL-приложений: от простейших презентаций до браузерных игр. Достаточно вбить в поисковике соответствующий запрос, чтобы в ответ получить обширный список различного инструментария. И все же разработчики Blend4Web нашли, чем удивить требовательного современного пользователя.

Blend4Web — это аббревиатура от нескольких слов: *Blender*, *Web* и *Four*. Учитывая созвучность слова *four* [англ. четыре] со словом *for* [для], получается следующий перевод: *Blender* для *Web*.

Blender — истая жемчужина мира Open Source, мощный и удобный редактор для 3D-моделирования. Его возможности широки: от создания игровых моделей до кинематографических эффектов. Однако отечественные программисты пошли еще дальше и добавили к *Blender* возможность экспорта сцены в WebGL. В итоге получилась полноценная среда для разработки



» Рис. 1. Универсальный плеер Blend4Web.

интерактивных презентаций. Давайте посмотрим, на что способен Blend4Web.

На сайте разработчиков можно скачать два варианта движка: плагин к *Blender* и полное SDK. Выбор зависит от того, что вы хотите сделать. Дело в том, что разработчики в первую очередь позиционируют

необходимые зависимости для автономного запуска. Конечно, есть и более сложные варианты.

» Автоматически в HTML внедряется плеер для проигрывания презентации.

Особенностью Blend4Web является возможность использования готового плеера. Он удобен и предлагает хороший набор функций прямо «из коробки». Вы можете рассматривать модель со всех сто-

«Без строчки кода пользователь должен получить то, что желает.»

Blend4Web как систему для дизайнеров. Именно им пригодится версия плагина. Без строчки кода, только путем настройки параметров в панелях *Blender*, пользователь должен получить то, что желает. Если встроенных функций не хватает, то на помощь приходит возможность скриптования на JavaScript и, соответственно, использование SDK.

Основное назначение Blend4Web — это легкое создание виртуальных презентаций. Именно на этом сконцентрировано внимание разработчиков. Буквально за пару-тройку щелчков мыши можно получить из обычной сцены *Blender* полноценную «живую» презентацию, не внося никаких изменений.

Вот причины такого чуда:

» Blend4Web «понимает» все стандартные настройки сцены, материалов, анимации и даже физики.

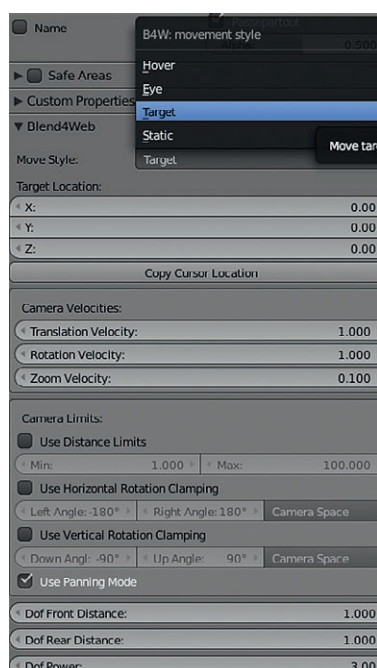
» Экспорт выполняется в единственный файл HTML, в котором содержатся все

необходимые зависимости для автономного запуска. Конечно, есть и более сложные варианты.

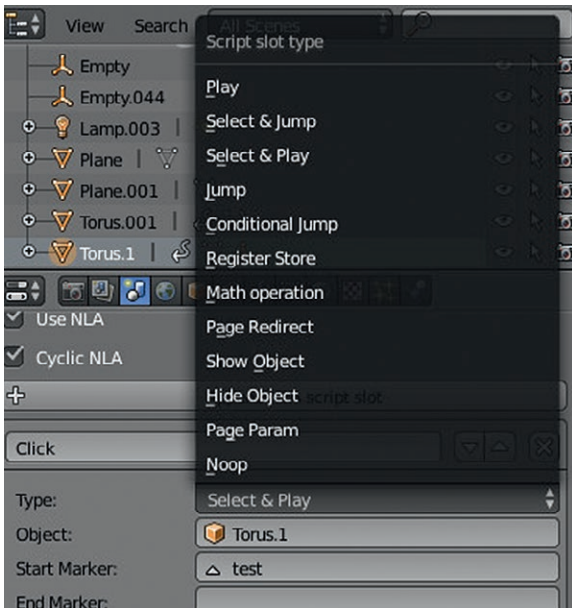
Однако если бы этим все ограничивалось, то будущее движка было бы незавидным. К счастью, это не так. Возьмем, к примеру, поведение камеры.

По умолчанию она вращается вокруг центра сцены (координаты XYZ: 0, 0, 0). Открыв панель настроек камеры, вы увидите новую вкладку с названием Blend4Web (рис. 2). Здесь имеется большое количество опций для корректировки: привязки к другим координатам XYZ, установке ограничений движения и даже фокусного расстояния. А можно из выпадающего меню выбрать иной вариант поведения (см. рис. 2).

Закладки Blend4Web можно увидеть практически в любых панелях *Blender* и для всех объектов. Следует сказать, что новые опции дополняют, но не переписывают исходные. Так что бояться испортить



» Рис. 2. Дополнительные настройки камеры от Blend4Web.



► Рис. 3. Здесь настраивается логика.

проект не приходится. Все они работают только после экспорта.

Blend4Web прекрасно понимает базовые виды анимации *Blender*: скелетную, объектную и даже вершинную. Причем вы можете полноценно использовать NLA Editor для управления последовательностью проигрывания. Таким образом создается несложный презентационный ролик или мультфильм. А вот для настоящей интерактивности разработчики придумали так называемые слоты NLA.

Если вы знакомы с *Blender*, то знаете, что в своем составе он имеет полноценный игровой движок Blender Game Engine. Для построения логики в нем используется визуальный редактор. Он удобный и функциональный. Логично было бы предположить его использование и для Blend4Web. Но разработчики пошли своим путем.

Слоты NLA — это дополнительные поля в панели Scene, которые позволяют настроить достаточно примитивную логику приложения. Не рассчитывайте с их помощью создать игру. Хотя для презентации вполне достаточно (рис. 3).

Давайте рассмотрим, как все это работает. Допустим в сцене имеется два объекта, один из которых должен проиграть анимацию после выделения другого. Соответственно, понадобятся два слота:

- **Select & Jump** Указываем в настройках первый объект, после выделения которого произойдет переход на слот 2.
- **Play** Проигрываем анимацию второго объекта.

Просто, действительно просто. Но тут-то пользователя и поджидает неприятный сюрприз. Слоты в панели выстраиваются друг за другом, и нет возможности визуально проследить иерархию. В сложных построениях вполне реально запутаться, что очень ограничивает их использование.

Да и набор возможных слотов небольшой. Список исчерпывается условными и безусловными переходами, управлением видимостью объектов, простыми математическими операциями и своеобразными регистрами памяти для сохранения результатов вычисления.

Несмотря на скудный набор для построения логики, его достаточно для реализации презентации. Ведь дополнительно к ней можно использовать специальные функции Blend4Web, которые позволяют решать наиболее популярные задачи. Например, отслеживание выделения объекта в сцене мышью или тактично при использовании мобильного устройства. Включаете соответствующие опции для нужного объекта, выбираете цвет заливки, режим прорисовки и устанавливаете слоты. Вот и все, и никаких скриптов.

Заслуживает внимание простая система интерактивной справки, которую вы можете использовать для подписей к модели. Добавьте объект Empty [Пустышка], заполните поля с кратким и полным описанием в панели — и после экспорта в сцене появятся симпатичные ярлычки с надписями. Если щелкнуть по ним мышью, то откроется само описание (рис. 4).

Эти два примера показывают, что Blend4Web позволяет быстро и легко создавать

частью является физическая модель поведения машины. Blend4Web предлагает для этого воспользоваться специальной заготовкой. Достаточно совместить колесную систему с опциями в специальной панели — и машина поехала.

Нужен SkyBox? Есть замечательный процедурный SkyDom. Ветер, реалистичная вода с пеной, к ней же физика плавающих объектов, постпроцессинговые эффекты и многое, многое другое. Достаточно открыть справку по движку. Кстати, она составлена на русском языке (имеется и английский эквивалент).

Наступает время подводить итоги. Сначала мы рассмотрим положительные моменты:

- **Интеграция с Blender** Движок понимает практически все настройки материалов, анимации, физики, окружения и частиц.
- **Наличие заготовок** Наборы для создания презентаций, физика транспортных и плавающих средств, настройка динамического окружения.
- **Мощная визуальная часть** Поддержка всех типов текстур, постпроцессинговые эффекты, нодовые материалы, освещение и тени.
- **Быстрый экспорт в HTML** и компактный размер готового файла

Конечно, не обошлось и без некоторых минусов:

- слабая визуальная система программирования;
- несколько запутанное расположение настроек движка (в недавно вышедшей версии 15.06 разработчики отчасти решили эту проблему);
- документация к движку не всегда поспевает за релизами.

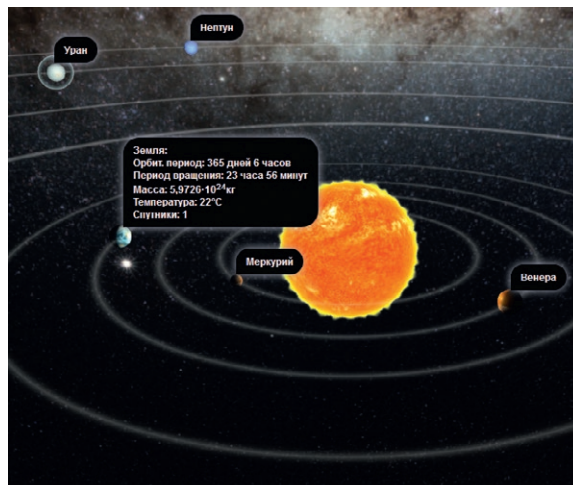
В целом можно сказать, что это динамично развивающаяся, перспективная разработка (релизы каждый месяц), которая полностью готова к использованию для трехмерных презентаций. Что же касается создания игр, то тут не всё так однозначно. Но попробуйте и расскажите. **LXF**

«Позволяет быстро и легко создавать визуальные презентации.»

виртуальные презентации. Но в действительности возможностей движка гораздо больше, и при использовании скриптов уже можно задумываться о создании на нем игр.

Разработчики Blend4Web стараются облегчить жизнь пользователям, подготавливая заготовки для разных случаев. Например, при создании гонки сложнейшей

► Рис. 4 Интерактивная справка в действии.



LINUX FORMAT **Вердикт**

Blend4Web

Разработчик: ООО «Триумф»
 Сайт: www.blend4web.com
 Цена: Бесплатно под GPL + коммерческая лицензия

Функциональность	8/10
Производительность	8/10
Простота в работе	8/10
Документация	9/10

► Перспективная, динамично развивающаяся система. Требует знания Blender.

Рейтинг 8/10

Shadowrun Chronicles: Boston Lockdown

Евгений Балдин решил написать хронику неудачи игры на основе интересной в целом вселенной.

Вкратце

» Тактическая пошаговая стратегия с упором на совместное прохождение кампании.



» Маг хорош против толп. К сожалению, противники это тоже знают, поэтому быстро разбегаются и прячутся.

Вселенная *Shadowrun* — это в каком-то смысле синтез фантазий профессора Толкиена и классической «твёрдой» научной фантастики. Это мир, где драконы активно занимают

программированием, а заговор от сглаза — это не блажь, а суровая необходимость. В последние два года в этой вселенной наступил небольшой ренессанс. Основным «движителем» повышения

интереса к ней стала американская независимая студия Harebrained Schemes, которая выкупила права на вселенную у Microsoft и выпустила на её основе пошаговую однопользовательскую тактику с элементами ролевой игры и классического квеста *Shadowrun Returns* (\$1,8 млн на Kickstarter с 2012 г.) с действительно великолепным расширением *Shadowrun: Dragonfall* (шло как часть базовой кампании). Сейчас в Harebrained Schemes идёт работа над следующим дополнением *Shadowrun: Hong Kong*, на которую через Kickstarter уже в 2015 г. собрали свыше \$1,2 млн.

Shadowrun Chronicles делали совсем другие люди. Хотя они и пытались скопировать достаточно успешную модель поведения Harebrained Schemes, но что-то пошло не так. Попробуем разобраться, что именно.

Игру взялась разрабатывать относительно небольшая (около 25 человек личного состава) австрийская игровая

Свойства навскидку



Выбор оружия

Куклу-персонажа можно одевать; дополнительные одежды можно получить при прохождении сюжетных миссий.



Выбор команды

Риггер, декер, маг и стрелок. Эта команда получилась не очень сбалансированной. Не хватает «автоматчика».



➤ История получилась довольно нетривиальной. Хотелось бы почитать другие части этой серии.



➤ Даже декера можно защитить магическим защитным полем. Целиться ему это не мешает.

студия Cliffhanger Productions. Студия заявила о своём проекте в 2012 г. на волне роста популярности финансирования игр на Kickstarter, с планами выпустить её в конце 2013 г.. Хотели они на свой проект *Shadowrun Online* всего-навсего полмиллиона долларов. Они, хоть и с некоторым трудом, эту сумму собрали. Значительную роль, скорее всего, при этом сыграла словесная поддержка ключевых фигур из Harebrained Schemes и довольно быстрое осознание того, что за схему "free to play" никто на Kickstarter платить не будет. Перед официальным выходом игры, которое произошло в мае 2015 г., было даже изменено название на *Shadowrun Chronicles: Boston Lockdown*, чтобы даже ассоциации с мобильными развлечениями не осталось.

Студия отметилась относительно популярной пошаговой тактической игрой *AErena*, ориентированной на мобильные платформы и дуэли. Поэтому не удивительно, что *Shadowrun Chronicles* тоже

➤ Игра начинается с того, что ГГ просыпается и ничего не помнит. Позже выяснится, что в этом есть смысл.

привязали к своему серверу. А значит, без интернет-соединения не пройти даже одиночную кампанию. В принципе, для крупных городов это давно не проблема,

«Игра очевидно поощряет социализацию — если та вам нужна.»

но когда после окончания бета-тестирования были «открыты шлюзы» для всех желающих, случилась катастрофа. На игровой сервер сразу пришло много людей и они, все как один, начали играть...

На Metacritics в случае *Shadowrun Chronicles* не числится ни одного исключительно положительного критического обзора. Средний балл 63. И виной этому чудовищные задержки, которые достигали нескольких минут, а то и десятков минут между командой и её реальным исполнением. Что делают люди, когда после отдачи команды персонажи продолжают чесаться и ковырять в носу, а не бегут вприпрыжку к указанной позиции? Правильно: опять отдают команду, опять и опять, что приводит к ещё большей задержке, поскольку для обработки команд предусмотрен ровно один центр,



отчаянно «тормозящий». Разработчики довольно быстро нашли виновника: это был «сборщик мусора», но игрокам легче от этого не стало. Прошёл месяц до момента, когда разработчики смогли укротить свой мусоросборщик, да и ажиотаж ожидаемо спал. Cliffhanger Productions подала на банкротство, и хотя её дочерняя структура Cliffhanger Games, обслуживающая сервис *Shadowrun Chronicles*, не подпадает под него, вряд ли будущее этого проекта будет радужным, а жаль.

На самом деле, если вычеркнуть каким-то образом воспоминания о лагах, игра вполне имеет право на существование. Хорошая, хоть и несколько упрощённая по сравнению с тем же *Shadowrun Returns* боевая составляющая. Риггеры, контактные бойцы, штурмовики, маги и декеры получились достаточно разными, чтобы набирать совместную команду. У мага самая дешёвая атака по площади, зато декер может поражать цель через стену. Автоматчики владеют самым дальнобойным оружием, а риггеры, особенно в начале, пока их не ослабили, просто раскатывали противников гусеницами своих боевых платформ. Вполне достойная история с большим количеством не обязательного для прочтения текста,

который создаёт вокруг ощущение живого, но и фантастически-фантазийного Бостона.

Относительно небольшие арены делают сражения достаточно динамичными и не затянутыми. Искусственный интеллект острой уму особо не блещет, но и не допускает глупых ошибок. Разнообразие экипировки и одежды позволяет одевать персонажа по-новому перед каждым выходом в «свет». Игра очевидно поощряет социализацию — если та вам нужна, конечно... **LXF**

LINUX FORMAT Вердикт

Shadowrun Chronicles: Boston Lockdown

Разработчик: Cliffhanger Productions
Сайт: shadowrun.com/shadowrun-online
Цена: 499 руб. в Steam

Сюжет	7/10
Графика	7/10
Увлекательность	6/10
Оправданность цены	4/10

➤ Интересная вселенная, нормальная история, затягивающие бои, но центральный сервер всё портит.

Рейтинг **6/10**

Dell XPS 13 Dev Ed

Наконец-то появился ноутбук с Linux — на зависть владельцам MacBook Air и к радости Хуана Мартинеса.

Вкратце

» Чудесный Linux (Ubuntu) в чудесном ноутбуке. Хорошо сконструированный, тонкий и легкий 11,9-дюймовый ноутбук с 13,3-дюймовым HD+ экраном и достойной долговечностью батарей.

Спецификация

- » ОС Ubuntu 14.04 LTS
- » CPU 2,4 ГГц Intel Core i7-5500U
- » GPU Intel HD Graphics 5500
- » ОЗУ 8 ГБ DDR3
- » Дисплей 3200×1800, сенсорный UltraSharp QHD+
- » Диск 512 ГБ SSD
- » Порты 2×USB 3.0, mini DisplayPort, читалка 3-в-1 (SD, SDHC, SDXC)
- » Связь 802.11ac, Bluetooth 4.0, web-камера HD
- » Габариты 304×200×15 мм
- » Вес 1,34 кг

Хотите приобрести мощный, портативный и привлекательный ноутбук? Тогда главным претендентом в сфере Linux будет Dell XPS 13 Developers Edition. У него изящный дизайн, легкие шасси и революционный дисплей, который заставит производителей пересмотреть способы изготовления ноутбуков.

XPS 13 встроено в серебристый алюминиевый корпус, похожий на MacBook Air по блеску и прочности. Это превосходный материал: после недели тестирования на нем не осталось ни отпечатков пальцев, ни царапин, ни пятен. Dell сделал опору для рук и ложе мелкоклавишной клавиатуры из изысканного углеволокнистого композита, приятного на ощупь. В отличие от некоторых ноутбуков, чья опора для рук сделана из того же материала, что и корпус устройства, т.е. из пластика или алюминия, опора для рук XPS не перегревается и не потеет.

Истинным чудом XPS 13 является его соотношение экрана к панели. Вы получаете экран 13,3 дюймового (338 мм) ноутбука на устройстве размером 11,9 дюймов (304 мм).

В Dell умудрились создать дисплей, занимающий 80% панели (на 11% большая площадь экрана, чем у 13-дюймового MacBook Air). Они сумели растянуть экран на несколько сантиметров в верхних углах и примерно на дюйм в нижней части передней панели. Эта функция создает эффект просмотра на большом экране и делает сенсорный экран куда удобнее. Дисплей стандарта Quad HD+ предлагает разрешение 3200×1800 — а большинство

» Невероятно! Dell втиснул 13,3-дюймовый Quad HD+ дисплей в 11,9-дюймовые шасси.



13-дюймовых дисплеев не выходят за рамки 1920×1080.

В дополнение к повышенной четкости изображения XPS 13 не слепит блеском и предлагает прекрасные углы обзора, облегчающие групповую просмотр.

Прелести Ubuntu

У чудесного экрана XPS 13 есть только два недостатка: во-первых, Dell пришлось разместить web-камеру на нижней левой стороне лицевой панели; во-вторых — для родителей маленьких детей или тех, кто переносит свои устройства вручную — XPS 13 нельзя брать за верхние углы панели. В целом, с точки зрения размера, ноутбук крошечный: по сравнению с MacBook Air он на 0,8 дюйма короче, на дюйм тоньше и на 1,1 дюйма меньше в высоту. И это одно из самых легких устройств на рынке ноутбуков.

Во время нашего теста на долговечность батарея XPS 13 продержалась 4 часа и 21 минуту, выполняя одновременно самые разные задачи при максимальной яркости экрана. Установив стандартную яркость, 50%, мы получили 7 часов 40 минут воспроизведения видео, что весьма конкурентоспособно, особенно при разрешении QHD+.

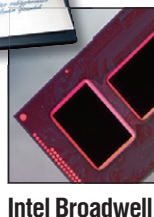
XPS 13 идет с (теоретически) крутой кнопкой заряда батарей, расположенной на левой стороне основания. Нажав на эту кнопку, вы увидите 0–5 лампочек, обозначающих процент заряда. Проблема с этой функцией в том, что на поиск крошечной кнопочки уходит столько усилий, что куда логичнее просто открыть ноутбук и проверить заряд на экране. Еще один огрех дизайна, который хотелось бы

увидеть исправленным в следующем году — расположение кнопки питания XPS. Сейчас она опасно близка к кнопке Delete. И мы пару раз случайно на Delete нажали, из-за чего наша работа приостановилась. Кроме того, мы обнаружили, что тачпад сверхчувствителен к касаниям: наш курсор то и дело отправлялся в какие-то загадочные уголки рабочего стола, а окна браузера открывались и закрывались.

Блестяще разработанный ноутбук Dell XPS 13 превосходит конкурентов по производительности, а дисплей Infinity окажет влияние на дизайн ноутбуков в сторону значительного увеличения удобства для пользователей в работе с их устройством. К счастью, XPS 13 отнюдь не является примером красоты без мозгов, и обладает достаточной мощностью, чтобы превратить работу и игру в удовольствие, а также достаточно долгим временем работы батарей, чтобы у вас не возникло проблем. **LXF**



Свойства навскидку



Intel Broadwell

Благодаря новейшей технологии Intel Core i7, с процессором 14 нм, мощность рассеяние CPU очень мала: 15 Вт.



Her Ethernet

Он настолько тонкий, что здесь нет порта Ethernet, но предусмотрен адаптер USB 3.0, если кому надо.

LINUX FORMAT Вердикт

Dell XPS 13 Developers Ed

Изготовитель: Dell
Сайт: www.dell.com/uk
Цена: £ 1235

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

» Не стоит больше завидовать владельцам MacBook Air, поскольку Dell XPS 13 — один из лучших ноутбуков.

Рейтинг **9/10**

Asus C300M

Красочный хромбук сбалансировал цену и производительность, чего, по мнению **Карла Ходжа**, вполне достаточно, чтобы его полюбить.

Спецификации

- » **CPU** Двухъядерный Intel Celeron N2830 2,16 ГГц
- » **GPU** Intel HD
- » **ОЗУ** 2 ГБ DDR3
- » **Дисплей** 13,3 дюйма 16:9 HD (1366×768 пикселей)
- » **Диск** 32 ГБ SSD
- » **Порты** 1×USB 3.0, 1×USB 2.0, HDMI, разъем SD-карты, вход для микрофона и наушников
- » **Связь** Wi-Fi 802.11a/b/g/n/ac, Bluetooth 4.0, web-камера HD
- » **Габариты** 339×203×23 мм
- » **Вес** 1,4 кг

Кто бы мог подумать? Хромбук повзрослел. Ранние обзоры этих соединенных с облаком легко-весных ноутбуков отличались высокомерным тоном, однако Samsung Series 3 в конце 2012 г. положил конец этой критике. Пошла новая волна более крупных, лучших и более дерзких устройств — по-прежнему недорогих и портативных, но более мощных. Asus C300M — именно такое устройство. Из-за двухъядерного CPU Intel N2380 его характеристики довольно низки для ноутбука, зато вычислительная мощность выше, чем у большинства других хромбуков. Со своим тонким корпусом высотой 20,3 мм от крышки до дна, не имеющий вентиляторов C300M объединяет стиль ультрабука с мобильностью хромбука. 13,3-дюймовый экран покоряется тенденции увеличения размеров дисплея.

Asus C300M ощущается как хромбук: его крышка не достигает и 5 мм толщины, а корпус местами прогибается. Сборка весьма прочная, и он выглядит совершенно как стандартный ноутбук, пока вы не возьмете его в руки. Скругленные углы, характерные для хромбуков, добавляют дизайну мягкости. Проигрыш в прочности компенсирует портативность — он весит всего 1,4 кг. Это означает, что вполне возможно нести его открытым в одной руке и держать в другой чашку чая.

По части портов, C300M одних хромбуков лучше, других хуже: внизу есть порты HDMI, USB 3.0 и SD-карты. Имеется также дуальный аудиопорт. Родное разрешение — 1366×768, и всё смотрится четко и ясно, хотя и при относительно узком угле обзора.

Мы использовали Asus C300M две недели в качестве своей рабочей машины.

» **Asus C300M** сочетает стиль ультрабука с мобильностью хромбука.



Он путешествовал вместе с нами, мотался по поездкам и сидел на собраниях. Дополнительная ширина позволяет печатать на нем с большим удобством, чем на обычной портативной машине, хромбуке Samsung Series 3. Знакомая компактная [chiclet] клавиатура весьма функциональна.

Мощь и производительность

Google Apps, понятное дело, работают идеально. Имея всего-навсего 2 ГБ ОЗУ, хромбук начал тормозить только при запуске нескольких экземпляров браузера с несколькими открытыми вкладками в каждом.

Настоящее испытание для хромбука — потоковое воспроизведение мультимедиа. Воспроизводя «Темную сторону Луны [Dark Side of the Moon]» от Spotify Premium, мы смогли уловить большую часть деталей. Особого искажения не наблюдалось, однако звук был ненасыщенным и сосредоточенным на высоких частотах.

Потом мы открыли Amazon Instant Video, чтобы посмотреть потоковое HD-видео с разрешением 720p. Специальный выпуск Доктора Кто «Снеговика» воспроизводился без задержек при нашем Wi-Fi-соединении 802.11n — но именно здесь начали проявляться слабости ЖК-дисплея TN. И последним испытанием производительности в области мультимедиа стала браузерная игра *Bastion*. При полном разрешении C300M восхитительно справился с этой игрой в мультяшном стиле *Zelda*, причем *Chrome* редко демонстрировал падение скорости ниже 51 кадра в секунду.

Asus посулил длительность работы батарей 10 часов, и хотя в действительности мы ее не увидели, реальная длительность все же была довольно близка к заявленной. Рекордным был случай смешанного применения в течение 9 часов 16 минут, при настройке энергопотребления Keep Awake, чтобы экран не гасился. И это выдающаяся производительность по любым меркам.

Мы бы, разумеется, с большим удовольствием попробовали версию C300M с 4 ГБ ОЗУ — которые были бы под стать SSD-накопителю на 32 ГБ, но наше общее впечатление было очень благоприятным. Asus C300M крупнее прежних хромбуков, но остается очень мобильным компьютером, и его смело можно порекомендовать в качестве компаньона в деловой поездке. **LXF**

» Ну что, подъем? Цвет у Asus чрезвычайно яркий!



LINUX FORMAT Вердикт

Asus C300M

Производитель: Asus
Сайт: www.asus.com/uk
Цена: £219

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

» Со столь живучими батареями, это легковесный ноутбук для тех, кто все время на ходу. Если им нравятся яркие цвета.

Рейтинг 8/10

Chromebook Pixel

Pixel 2015 г. от Google — самый мощный хромбук на планете, но стоит ли ради него расставаться с такими деньжищами, призадумался **Кевин Ли**.

Спецификация

- » **CPU** 2,2 ГГц Intel Core i5-5200U
- » **GPU** Intel HD Graphics 5500
- » **ОЗУ** 8 ГБ DDR3
- » **Дисплей** 12,85 дюймов, сенсорный дисплей IPS 2560×1700
- » **Диск SSD** 32 ГБ
- » **Порты** 2×USB-C, 2×USB 2.0, разъем для наушников/микрофона, читалка SD-карт
- » **Связь** Intel Dual band Wireless-AC 7260; Bluetooth 4.0 LE; веб-камера 720p HD
- » **Габариты** 297×224×15 мм
- » **Вес** 1,5 кг

Первый Chromebook Pixel печально прославился тем, что это был самый дорогой хромбук с безумными спецификациями, оснащенный неортодоксальным экраном 3:2 с высоким разрешением. Это было дорого и шикарно, однако время заряда батареи оставляло желать лучшего. Новый Pixel оснастили таким же экраном с разрешением 2,560×1,700 и даже более мощным процессором Intel Core i5 Broadwell, плюс к тому заявленная продолжительность заряда 12 часов — причём всё это ещё и по более низкой цене. Во множестве отношений, новый Pixel — это усовершенствованный хромбук.

Он было напомнил нам автомобиль DeLorean, и не в хорошем смысле, показавшись кустарным и унылым, со своим корпусом из листового металла в стиле машин 1980-х. К счастью, на деле вышло не так. В ваших руках — цельная алюминиевая пластина, напоминающая MacBook Air, но сделанная из более гладкого и чуть более мягкого металла. Откройте этот ноутбук, и он покажет вам строгий, элегантный и практичный дизайн. От нижнего края вокруг клавиатуры до самих клавиш и краёв трекада, ноутбук придерживается четкой эстетики прямых углов, заданной экраном 3:2. Края машины отвесные, что придает ей модернистский вид.

Правильный USB-C

Одно из важнейших новшеств Chromebook Pixel — новые порты USB-C. Новый разъем USB 3.1 немаловажен, в свете его



» Chromebook Pixel (2015) от Google — реальная мощь.

дебюта в новом MacBook. Однако, в отличие от последнего ноутбука от Apple, Pixel идёт с двумя новомодными портами вместо одного.

Два разъёма расположены по сторонам ноутбука, позволяя подключить зарядное устройство с любой стороны машинки. В то же время, наличие традиционных портов означает, что вам не надо облепляться адаптерами, как с новым MacBook. Не так уж много хромбуков, оснащенных HD-экраном, не говоря уже о 32 ГБ SSD или 8 ГБ ОЗУ по умолчанию. Это, безусловно, один из самых быстрых хромбуков, и он не тормозит даже после открытия 20 или более вкладок в *Google Chrome*. Работая на процессоре Core i5, машина отличается, воспроизводя потоковое вещание с качеством 1080p и даже видео 4K на YouTube. Большинство хромбуков хорошо справляются с FullHD, но более высокое разрешение заставляет большинство машин прихрамывать; Pixel же работает плавно.

Если абстрагироваться от неудобств, вызванных экраном 3:2, у Chromebook Pixel один из самых милых дисплеев на рынке. Панель приятная и яркая: даже и при подсветке, выставленной на 50%, всё происходящее на экране прекрасно видно, а яркий солнечный свет не вызывает проблем. Цвета насыщенные, и контрастности достаточно, чтобы в полной мере оценить переход белого в мягко-серый и затем — в глубокий чёрный. В Google даже отрегулировали отзывчивость и точность тачскрина. Мы, правда, нечасто ловили себя на игре с ним, кроме разве что

растягивания щипком для увеличения изображения на некоторых сайтах.

Google счел возможным похвастаться новой жизнью батареи Chromebook Pixel в 12 часов, и это недалеко ушло от истины. Во время тестирования мы добились 8 часов и 22 минут интенсивного использования. Почти 9 часов работы батареи — это уже огромное время для любого хромбука, и у Pixel оно ещё более впечатляет — тем более при экране с высоким разрешением и процессоре Core i5.

В Google улучшили Pixel почти во всех отношениях, а главное, решили вопрос с батареей. Новый Pixel также украшен более насыщенным и отзывчивым тачскрином и добавлением USB-C; и если цена вас не смутила, эту модель приобрести обязательно следует. **LXF**



Свойства навскидку



USB-C

В целом, новые разъемы USB-C — это скорее благо для ноутбука, чем недостаток.



Минимализм

Мы надеемся, что вы любите аскетичный дизайн и клавиатуру с подсветкой, как мы.

LINUX FORMAT Вердикт

Chromebook Pixel (2015)

Изготовитель: Google
Сайт: www.google.com/chrome/devices
Цена: £ 799

Функциональность	8/10
Производительность	9/10
Удобство в работе	10/10
Оправданность цены	8/10

» Pixel — самый быстрый и хорошо продуманный Chromebook, но нужна ли вам эта мощь?

Рейтинг 9/10

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!



НОВОСТИ Дорогу — Android » iOS обходит? » Телефон слишком много о вас знал
» Lollipop помаленьку распространяется

МОБИЛЬНОСТЬ

Hyundai выбирает платформу

Android Auto повысит безопасность автомобиля.

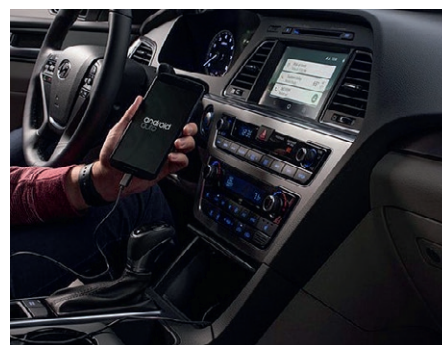
Компания Hyundai уже в текущем году начнет использовать платформу Android Auto в своих автомобилях и, таким образом, станет первым производителем, в автомобилях которого информация с подключенного Android-смартфона будет зеркально отображаться на дисплее информационно-развлекательной системы (ИРС). Android Auto с возможностями навигации был впервые показан в модели Sonata 2015 г., и позднее появится на других моделях Hyundai. Ранее Hyundai также заявила, что в Sonata 2015 опционально будет предложена и платформа Apple CarPlay.

В компании Google, создателе Android Auto, полагают, что в конечном итоге транспортные средства превратятся в еще одну платформу для мобильных устройств, поскольку все чаще и чаще имеют изначально встроенные в них мобильные операционные системы.

Впервые объявленная в июне 2014 г., платформа Android Auto (совместимая с версиями 5.0 "Lollipop" и выше) поддержана организацией Open Automotive Alliance, включающей Audi, General Motors, Honda и Hyundai, а также Google и еще рядом

технологических компаний. Конкурентами Android Auto являются Apple CarPlay и такие открытые стандарты, как MirrorLink или GENIVI. Все они допускают использование мобильных приложений (таких как Google Maps или iTunes) с отображением на дисплее ИРС автомобиля.

Год назад объявила о своем альянсе с Android Auto и компания Hyundai. «Android Auto отлично сочетается с основными принципами дизайна интерьера Hyundai по безопасности, интуитивности и простоте, — заявляет CEO Hyundai Motor America Дэйв Зуховски [Dave Zuchowski]. — Мы выполнили своё обещание, добавив эту долгожданную функцию в пользующейся высоким спросом модели Sonata. Отображение Android OS на дисплее ИРС не только предлагает навигационные обновления в реальном времени, поиск и развлечения, но и повышает безопасность, так как нет необходимости смотреть на смартфон в процессе движения. Взгляд и внимание водителя будут сосредоточены на дороге, поскольку функции смартфона интегрированы с центральным дисплеем, физическими элементами управления и микрофоном автомобиля. Кроме того, экран



» Hyundai Sonata станет первым автомобилем, изначально располагающим Android Auto.

смартфона блокируется, и у водителя нет необходимости отвлекаться на свой телефон».

По информации Hyundai, 660 000 водителей используют сотовые телефоны или манипулируют электронными устройствами во время вождения. Эта цифра стабильна с 2010 г.

Среди сервисов Microsoft, которые будут интегрированы в Cyanogen OS, названы Bing, Skype, OneDrive, OneNote, Outlook и Microsoft Office.

ДОЛИ РЫНКА

Все больше Lollipop

Сейчас более 10% устройств на Android используют эту версию.

Согласно данным, приводимым Google, доля Lollipop на рынке возросла с 9,7% в начале мая до 12,4% к 1 июня. При этом на долю Android 5.0 приходится 11,6% устройств, тогда как новейшая версия мобильной платформы от Google, Lollipop 5.1, набрала всего лишь 0,8%. Рост, конечно, не стремительный, но устойчивый, а недавний выпуск Lollipop-устройств уровня top-end, таких как Samsung Galaxy S6 и LG G4, обещает эту тенденцию подкрепить.

Тем не менее, по сравнению с более старыми версиями Android, доля рынка Lollipop в целом по-прежнему мизерна. Так, Android 4.4 Kitkat всё еще используется на 39,2% смартфонов и планшетов на Android (небольшое снижение относительно

39,8% в мае), а доля Jelly Bean 1 июня составляла 37,4% (39,2% в мае). Еще более старые версии Android, Ice Cream Sandwich, Gingerbread и Froyo, установленные на 5,1%, 5,6% и 0,3% устройств соответственно.

Для сравнения, статистика использования дистрибутивов в Apple App Store дает иную, менее фрагментированную картину. По состоянию на 25 мая, система iOS 8 была установлена на 82% устройств (iPhone и iPad), в то время как iOS 7 — на 16% устройств. И всего 2% устройств используют более старые версии Apple iOS.

На конференции разработчиков I/O Google 2015 года была показана следующая итерация платформы Android. В порядке ответного хода,



» Более чем 10% Android-устройств используют платформу Android 5.0 Lollipop.

на конференции Worldwide Developer Conference компания Apple ожидается «приподнимет покровы» над iOS 9.

КОНФИДЕНЦИАЛЬНОСТЬ

Злопамятные смартфоны

Даже сброшенный к заводским настройкам Android-смартфон сохраняет данные пользователя.

Перед продажей или утилизацией мобильного устройства многие пользователи «сбрасывают» его к заводским настройкам. Однако тест 21 подержанного смартфона с версиями Android от 2.3.x (Gingerbread) до 4.3 (Jelly Bean) показал возможность восстановления, даже после сброса к заводским настройкам, электронных писем, текстовых сообщений, маркеров доступа Google и других конфиденциальных данных. Для исследования сотрудники Кембриджского университета Лоран Симон [Laurent Simon] и Росс Андерсон [Ross Anderson] использовали приобретенные на eBay в январе–мае 2014 г. смартфоны производства Samsung Electronics, HTC, LG Electronics и Motorola, плюс три модели линейки Google Nexus.

В 80% удалось восстановить ведущий маркер Google, что позволяет атакующему ресинхронизировать устройство с учетной записью Google предыдущего владельца, получив доступ к его электронным письмам, контактам, паролям Wi-Fi и другим сохраненным данным. В некоторых случаях были восстановлены маркеры доступа Facebook, части электронных писем, SMS-сообщений и других средств мгновенного обмена

сообщениями. «Причины проблем многообразны. Новые телефоны в этом плане, как правило, надежнее старых, а модели бренда Google надежнее, чем предложения OEM», написал Андерсон в блоге.

Шифрование снижает риски, но не защищает телефон полностью. В некоторых случаях методом «грубой силы [brute force]» удалось восстановить достаточное количество информации даже при поврежденном сбросом настроек ключе шифрования. Усложняет задачу надежный пароль, поскольку обычный 4-разрядный PIN взламывается достаточно быстро.

По оценкам исследователей, до 500 млн устройств могут не затирать должным образом раздел, на котором располагаются ОС и учетные данные приложений, и до 630 млн могут не санировать SD-карту, где обычно сохраняются мультимедиа-файлы.

Отдельно были проанализированы «противоугонные» функции, предлагаемые основными мобильными приложениями безопасности: удаленная блокировка устройства и стирание данных. Вывод таков: эти приложения не представляют собой альтернативы некорректно выполняемой



➤ Сброс телефона с Android к заводским настройкам не гарантирует надежного удаления конфиденциальных данных.

встроенной функции сброса к заводским настройкам, их возможности ограничены архитектурой ОС и API. Единственным эффективным решением было бы улучшение механизма стирания данных, сделанное производителями телефонов.

Пока не ясно, была ли улучшена функция возврата к заводским настройкам в версиях Android новее 4.3. На просьбу прокомментировать полученные результаты Google не ответил.

НАШИХ БЬЮТ?

Спад Android, рывок iOS

Производители стремятся внедрять собственные интерфейсы пользователя.

Тревожна для Google тенденция первого квартала (Q1): хотя продажи Android-смартфонов и увеличились, доля этой платформы на рынке продолжает снижаться. Показатели же iOS растут третий квартал подряд. В целом, по данным компании Gartner, за год рынок вырос на 19,3% — конечным пользователям во всем мире продано 336 млн смартфонов. При этом доля Android снизилась на 1,9 пункта, до 78,9%, а для iOS зафиксирован рост с 15,3% до 17,9%.

За изменениями в годовом исчислении скрывается еще более резкое падение Android: 83,1% в Q3 прошлого года, и стремительный взлет iOS с 12,7%, что вызвано началом продаж iPhone 6.

Ключевым полем битвы в первом квартале года был Китай: рекордные продажи iPhone поколебали позиции Android в этой стране. В годовом исчислении, согласно Gartner, впервые в Китае объемы Android упали на 4%. По прогнозам

IDC, для рынка смартфонов тенденция сохранится, и в этом году, опять же впервые, доля Android будет расти медленнее, чем общий объем продаж смартфонов.

У Google есть и другие причины для беспокойства, помимо замедления роста объема продаж. Сторонние производители — Lenovo, Huawei Technologies, Xiaomi и Micromax — используют интерфейсы, значительно отличающиеся от стоковой версии Android. Одновременно возрастает конкуренция с сервисами, которыми комплектуются их модели. С этой точки зрения неудивительна позиция компании Google, пытающейся расширить охват Android на автомобили, TV и т.п.

В Q1 борьба шла не только между Android и iOS, но и между Samsung Electronics и Apple. После того, как лавина продаж iPhone 6 в конце 2014 г. вывела Apple вперед, в Q1 Samsung снова занял первую строчку, продав 81,1 млн смартфонов по сравнению с 60,2 млн у Apple. Прибыль

Apple в последнем квартале, в значительной степени благодаря продажам iPhone, составила \$13,6 млрд.

До 8,3 млн увеличились продажи смартфонов на Windows Phone, однако доля этой платформы снизилась с 2,7 до 2,5%. Теперь надежды Microsoft связаны с Windows 10, которая предложит единый пользовательский интерфейс для устройств всех типов. Что же касается продуктов BlackBerry, то покупатели, похоже, их просто игнорируют: продано 1,3 млн смартфонов, доля на рынке 0,4% (в прошлогоднем Q1 — 1,7 млн устройств и 0,6%).

Поскольку функциональность и базовые возможности мобильных ОС все более и более сближаются, для разработчиков становится критически важным найти способы дифференцировать себя: расширить охват своей экосистемы; добавить новые функции; создать платформы, охватывающие несколько типов устройств. LXF

Сравнение

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

Работа на расстоянии

Какой клиент — самый лучший для полноценной деятельности в удаленном рабочем столе? Следствие ведет **Джонни Бидвелл**.



Про наш тест...

Для тестирования мы взяли высококлассный игровой ПК, Raspberry Pi 2 Model B (где есть программы), да еще сдули пыль с двудядерной машины 2,33 ГГц из Сравнения **LXF197**. Чтобы посмотреть результаты конкурсантов в сценариях с низкой пропускной способностью, мы пропустили клиенты через *Trickle*, программу формирования полосы пропускания. Так мы воссоздали низкую (25 кб/с) и очень низкую (6 кб/с) скорости соединения. Чтобы каждый клиент полностью реализовал свой потенциал, мы сначала соединили каждый клиент с его партнерским сервером для обеспечения максимума производительности. Мы даже взяли ПК с Windows 7, чтобы протестировать возможности клиентов RDP на фоне исходного протокола (сервер с открытым кодом *xrdp* реализует только части протокола на общественном домене). В разделе совместимости мы всё это смешали, для оценки взаимодействия разных клиентов и серверов.

Хотя всем известно, что наилучшим способом удаленного доступа является SSH, иногда было бы неплохо (а порой и необходимо) получить доступ ко всему рабочему столу. Может, вам надо научить свою тетушку Этель менять фон рабочего стола или заставить птар делать схему сети конкурирующего кружка вязания.

Этот избыток графических данных представляет собой проблему, особенно при отсутствии широкополосного соединения, и есть ряд технологий, призванных эту проблему решить. Linux отдает предпочтение протоколу VNC, а Windows — по большей части закрытому Remote

«Может, надо научить тетушку Этель сделать схему сети конкурирующего кружка по вязанию.»

Desktop Protocol (RDP). Однако ни один из протоколов не отдает предпочтения какой-то ОС. Оба работают напрямую с буфером кадров, поэтому лежащая в их основе технология одинаково годится и для Windows, и для Linux. Протокол NX, используемый в NoMachine NX, может составить им обоим конкуренцию благодаря своим трюкам со сжатием и сокращению запаздывания, которые в Linux работают

напрямую на протоколе X (а в Windows — на протоколе RDP). Однако с 2010 г. клиент имеет закрытый код, и хотя ряд проектов некогда намеревался обеспечить решения NX с открытым кодом, их разработка по большей части отошла на второй план, за исключением X2Go. Приложение *Chrome Remote Desktop* по-прежнему находится на стадии бета, однако для некоторых оно будет представлять интерес.

Наша подборка

- » Chrome Remote Desktop
- » NoMachineX
- » Remmina
- » TigerVNC
- » X2Go

Простота в работе

Удобны ли они в установке и навигации?

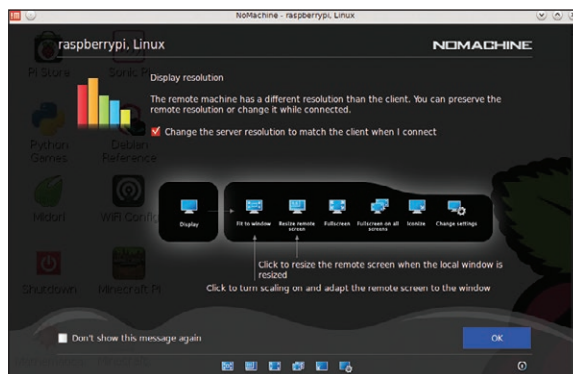
Вряд ли отыщется дистрибутив, не содержащий пакета *Remmina*, поскольку тот довольно популярен. Чтобы запустить функции VNC в *Remmina*, нужно установить *libvncserver*, однако большинство дистрибутивов решают эту проблему за вас. В Arch Linux этот пакет был перечислен среди опциональных зависимостей, и его придется устанавливать вручную. Несмотря на изобилие опций, все в *Remmina* организовано интуитивно, так что простое соединение просто и настраивается.

Зато обнаружить пакеты *TigerVNC* может быть весьма непросто. Многие дистрибутивы, включая Debian и Ubuntu, придерживаются более старого и отличающегося двумя буквами *TightVNC*. Но стоит вам пакеты добыть, как вы запросто найдете общий язык с этим клиентом. Он создан по образцу «исходного» клиента *RealVNC*, и по этой причине там особенно не на что смотреть. Опции по умолчанию просто работают, и соединение сводится к вводу имени хоста в адресную строку, хотя в конце, видимо, придется добавить :1.

Пакеты для *X2Go* есть у большинства дистрибутивов, включая Raspbian. После установки сервера вам, возможно, понадобится запустить `x2godadmin --createdb` перед соединением. Некоторые пакеты здесь, похоже, довольно ленивы. Клиент *Qt4* весьма прост в навигации, но, наверное, его структуру можно было сделать поаккуратнее. Он предлагает разумное управление сессией через список на правой стороне.

Установка браузерной части в приложении *Chrome Remote Desktop*, как легко догадаться, сложности не составляет. Однако настройка удаленного сервера (в Linux) включает установку Deb-пакета, который, судя по всему, не работает по умолчанию. В Mint/Ubuntu файлы надо перемещать или делать на них символические ссылки до того, как *Chrome* представит опцию включения удаленных соединений.

Это Сравнение посвящено удаленным клиентам рабочего стола, но нам бы хотелось зайти чуть дальше и подвергнуть это приложение порицанию за то, что



» Второй из четырех экранов мастера, который приветствует вас в мире *NoMachine*.

подобное неудобство неявно включено в работу с ним. Однако после того, как все будет настроено, получится простой для понимания список компьютеров и запросов на удаленную помощь.

NoMachine требуется устанавливать вручную, но там заготовлены пакеты Deb и RPM, а также пакет программы установки, если те вам не подойдут. Вы получите двойную порцию приветствий, что поможет вам освоиться; впрочем, интерфейс и без того достаточно простой (правда, на наш вкус несколько аляповатый). Серверы могут (опционально) показываться в сети, чтобы они были видимыми для всех клиентов.

Вердикт

NoMachine NX
★★★★★
Remmina
★★★★★
TigerVNCr
★★★★★
Chrome Remote Desktop
★★★★★
X2Go
★★★★★

» *Remmina* и *NoMachine* — самые дружелюбные из всех.

Документация

Вечно требуют прочитать руководство. А оно есть?

R*emmina* достаточно понятна и переведена на несколько языков. Если у вас достаточно храбрости, можете копнуть глубже, чтобы разобраться в работе команды `xfreerdp`, которая используется для сессий RDP. Несмотря на свой внешний вид, *TigerVNC* имеет

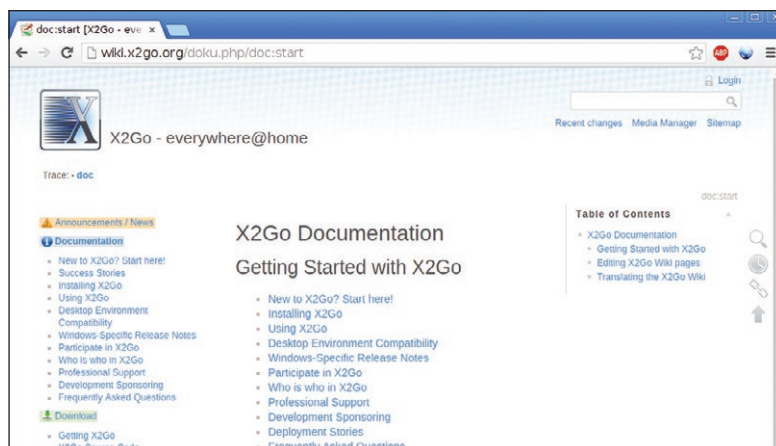
превосходные map-страницы. Он может быть особо интересен тем, кто хочет индивидуально настроить серверную сторону, но у этого клиента есть и опции командной строки для всего содержащегося в меню.

Сервер *X2Go* идет с набором утилит командной строки, удобных для скриптов.

Все они тщательно задокументированы на map-страницах. На сайте тоже немало полезной информации, не только по работе программы, но и относительно будущих идей проекта. Налицо также весьма удобный справочник, объясняющий, какие среды рабочего стола могут отказаться работать вместе и почему. Для простых случаев предложены решения, например, для *IceWM* и *OpenBox*, и весьма понятно изложены все плохие новости для всех желающих иметь современный рабочий стол.

Для настройки сервиса приложению *Chrome Remote Desktop* явно нужна документация получше. Да, оно пока на стадии бета, но это фундаментальная проблема. Или, еще лучше, почему бы не предложить рабочие пакеты? Само по себе приложение настолько простое, что даже тетушка Этель с ним справится.

Документация *NoMachine* более чем адекватная, но теряет баллы из-за показа четырех экранов с нудными инструкциями, прежде чем разрешить вам соединение.



» Сайт *X2Go* поможет вам приступить к работе, а их талисман вас очарует.

Вердикт

X2Go
★★★★★
NoMachine NX
★★★★★
Remmina
★★★★★
TigerVNCr
★★★★★
Chrome Remote Desktop
★★★★★

» Явный аутсайдер *X2Go* здесь, ко всеобщему удивлению, выигрывает.

Функциональность

У кого больше всех наворотов и финтифлюшек?

Все рассматриваемые клиенты, при надлежащей настройке, позволят вам удаленно соединиться с вашим рабочим столом.

Тем не менее они предназначены для разных целей, имеют разную направленность, и каждый располагает своими собственными хитростями. В данной категории мы рассмотрим, какие

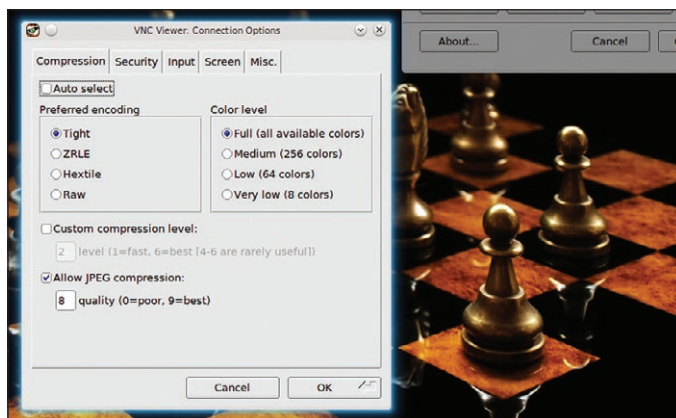
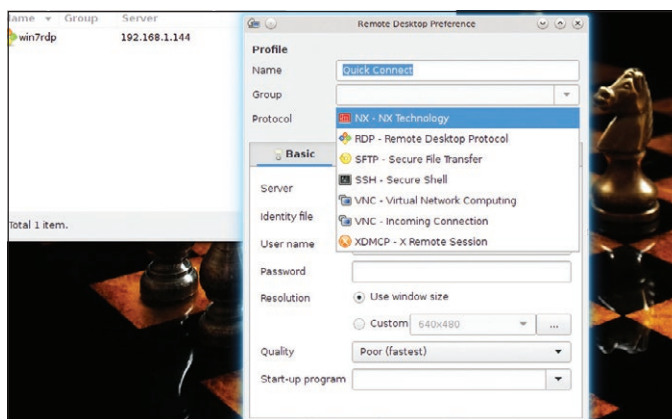
именно функции предлагает каждый кандидат, удобны ли они или оставляют пользователя в замешательстве.

Во всех тестируемых здесь клиентах поддерживается полноэкранный режим, так что (если скорость позволяет) вы можете притвориться, что сидите прямо перед удаленной машиной.

Некоторые упоминаемые здесь вещи являются исключительными свойствами клиента, а некоторые зависят от соединения клиента с нужным типом сервера. Последнее особенно относится к *TigerVNC*, так что многое из нашей похвалы или критики будет направлено скорее на серверную сторону.

Remmina ★★★★★

Remmina — впечатляющий клиент. Помимо NX, VNC и RDP, она поддерживает протокол XDMCP, который лежит в основе коммуникации между сервером отображения и менеджером отображения. Она даже разрешает соединения без графики через SSH и SFTP. Если вам этого недостаточно, ее можно расширить через систему плагинов. Она поддерживает предварительные настройки качества, которые вам надо будет изменить в зависимости от скорости соединения, а также опции масштабирования (если вы работаете на маломощном устройстве). Протокол VNC незашифрованный, и обычно соединение направляется через SSH. Его легко настроить с использованием опции SSH -L, но *Remmina* позволяет это сделать через диалоговое окно опций. Далее имеется буфер хранения с распределенным доступом, который работает на всех протоколах. Вы даже можете настроить клиент так, чтобы он слушал входящие соединения, чтобы помочь с проблемами брандмауэра.



TigerVNC ★★★★★

TigerVNC — пакет клиент/сервер, существующий с 1999 г. Он поддерживает несколько уровней кодирования и сжатия, так что наилучшие результаты могут быть получены даже при низкой скорости соединения. *TigerVNC* появился как ответвление практически усопшего *TightVNC*, и использует ту же стратегию деления экрана на четырехугольники (или даже шестиугольники) и использования наиболее подходящего сжатия (JPEG, zlib, всякие-разные мистические штуки) для каждого. Он остается совместимым с другими реализациями VNC, однако в них не видно предлагаемых им расширенных методов сжатия. У клиента спартанский интерфейс FLTK, который объединяет кнопки и адресную строку и имеет все виды кодирования и сжатия, опции безопасности (включая аутентификацию по сертификату TLS) и многое другое. Предлагается альтернативный сервер, *xVncserver*, для управления существующей сессией X вместо запуска нового.

Статус разработки

Есть ли у этих проектов будущее?

R*emmina* имеет право похвастаться плодотворной разработкой, не прекращавшейся с момента ее появления в 2009 г. Но это лишь интерфейс для *libvnc* и *xfreerdp*, ни один из которых не получает должного внимания помимо обычной поддержки. В результате, современный UX на GTK3 создает неверное представление о довольно устаревшей начинке. А ведь важна именно начинка.

Еретическое наследие исходного VNC (ныне *RealVNC*), *TigerVNC*, *TightVNC* и даже *TurboVNC* весьма непростое, но оказало

негативное влияние на реализации VNC с открытым кодом. *TigerVNC* находится в куда лучшей форме, чем остальные, но все же достоин большего внимания. *NoMachine*, напротив, похоже, неустанно прогрессирует. Жаль, что его код закрыт.

X2Go начинался в 2006 г. как проект двух студентов высшей школы. Сегодня поддержкой проекта занимаются они и основная команда из четырех человек. У них благородные цели, и это хорошо, поскольку как минимум 3 других проекта на базе протокола NX больше не поддерживаются.

Наш тест производительности показал, что *X2Go* пока несовместим с *NoMachine*, но последний показывает потенциал протокола и демонстрирует направление, куда надо двигаться команде.

О выходе Linux-порта *Chrome Remote Desktop* было объявлено в июле прошлого года, и его не самая блестящая производительность может быть простительна, поскольку работать в стиле NativeClient непросто; однако проект сумел предоставить пакеты, которые работают не только на Ubuntu 12.04.

Вердикт

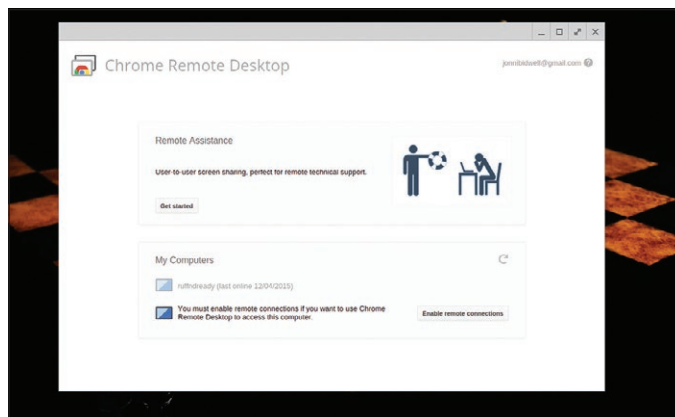
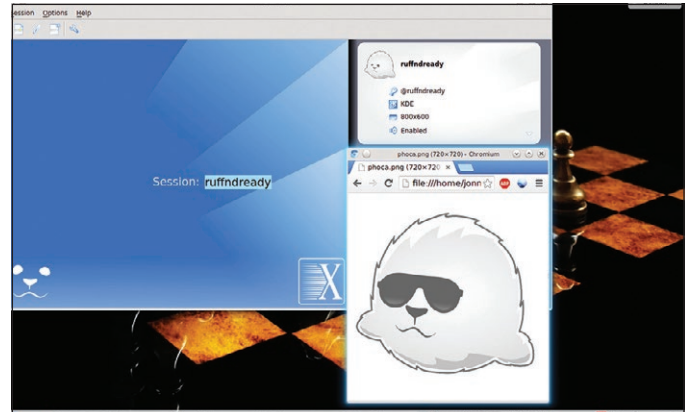
- NoMachine NXr ★★★★★
- Chrome Remote Desktop ★★★★★
- TigerVNC ★★★★★
- X2Go ★★★★★
- Remmina ★★★★★

» NoMachine трудится без устали, и жаждет погостить на все ваши машины.

X2Go ★★★★★

X2Go использует протокол NX над SSH. В нем много функций клиента *NoMachine*, но без проприетарных ужасов. Его милый талисман-тюлень зовут Фока [Phoca]. Кроме огромного количества схем сжатия, клиент позволяет выбрать скорость соединения.

Соединения со звуковыми системами, вроде *PulseAudio*, и файловыми системами можно направить через соединение SSH, чтобы обойти брандмауэр. Помимо обработки удаленного рабочего стола на отдельном дисплее, имеется также расширение *Xinerama*, которое позволяет справиться с рабочим столом на нескольких экранах. Поскольку сервер X2Go ответвился от старой версии X.org, он не поддерживает многих рабочих столов с ускорением 3D. В аварийном режиме Gnome 3, Unity и Cinnamon работать все-таки будут, но вам лучше попытаться с чем-нибудь попроще, например, с *Xfce* или *Mate*.



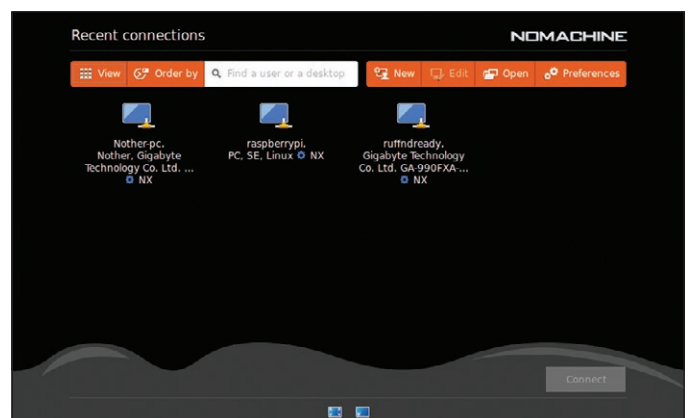
Chrome Remote Desktop ★★★★★

Кроме того, что он является удаленным клиентом рабочего стола, *Chrome Remote Desktop* больше нечем особо похвастаться. С другой стороны, возможность доступа к компьютеру с мобильного устройства бывает очень удобна — например, это может оказаться компьютер, на который вы не хотите устанавливать программы или не имеете соответствующих привилегий. Как только нужный сервисный пакет установлен на серверной машине (машинах), вы получаете комфортный доступ к ним из своего браузера. При условии, конечно, что вашим браузером является *Chrome* или *Chromium* и вы войдете в соответствующую учетную запись Google. Чтобы включить удаленные соединения на машине, вам сначала нужно задать PIN не менее чем из шести цифр. После соединения вы получаете лишь рудиментарный контроль за размером рабочего стола — приложение само решит, какое выбрать качество.

NoMachine NX ★★★★★

NoMachine NX выигрывает приз за крикливый интерфейс, благодаря своим большим черно-оранжевым окнам. Однако если вы простите ему это, как и его проприетарную натуру, то увидите, насколько он мощный. Как и *x0vncserver*, он дает доступ к работающему в данный момент рабочему столу, и подобно RDP, протокол NX поддерживает перенаправление аудио/USB/дисков. Он также разрешает передачу файлов и запись удаленной сессии рабочего стола. Только что вышедший пакет Raspberry Pi (который пока что в стадии альфа) тоже отлично работает.

NoMachine может использовать UDP для передачи мультимедиа-данных, которые он умеет кодировать с помощью сжатия H.264, VP8 или MJPEG. Последнее использует меньше энергии CPU и поэтому удобно для маломощных устройств. *NoMachine 4* больше не разрешает соединений SSH, но вы по-прежнему можете аутентифицироваться посредством публичного ключа.



0 протоколах и десктопах

Удастся ли взять другой протокол или любимый рабочий стол?

Хотя у *Remmina* самый разнообразный выбор протоколов, предлагаемая ею поддержка в основном хуже, чем у других клиентов. Большинство будут применять ее через VNC, что означает поддержку любого рабочего стола, но и отсутствие возможностей OpenGL на проприетарных драйверах. Использование ее с RDP не особо нас впечатлило, а плагин NX работает только для давно не рекомендуемых серверов *NoMachine 3*.

В *TigerVNC* не надо поддерживать другие протоколы, но опыт взаимодействия

с рабочим столом должен быть чуть приятнее, чем в *Remmina*, если вы не возитесь с настройками. Возможно, сервер придется запускать через **dbus-launch vncserver**, если компьютер сервера зарегистрирован в той же среде рабочего стола, что и нужная вам сессия VNC.

X2Go и прочие наши зверюги не слишком хорошо проявили себя в этой категории, потому что могут соединяться только со своей родней. Далее, поддержка рабочего стола тоже не слишком хороша: Qt5 не работает (так что KDE 5 у вас не будет),

Gnome и Unity не работают (остались проблемы, даже в аварийном режиме). Пока мы не установили KDE 4, мы сомневались, работает ли тут что-нибудь вообще.

Будем надеяться, что в скором времени проблемы с Qt5 будут исправлены в приложении *Chrome*. Но потом надо улучшать работу десктопа. Будет чудом, если заработает OpenGL, но до этого еще далеко. И хотя *NoMachine* может соединяться только со своей братией, его способность обслуживать даже самые сложные рабочие столы выглядит выдающейся.

Вердикт

- NoMachine NXr ★★★★★
- Remmina ★★★★★
- TigerVNC ★★★★★
- Chrome Remote Desktop ★★★★★
- X2Go ★★★★★

» Способность NoMachine работать где угодно выделяет его из общего ряда.

Выживание на низкой скорости

Будут ли они работать на скорости модемного доступа?

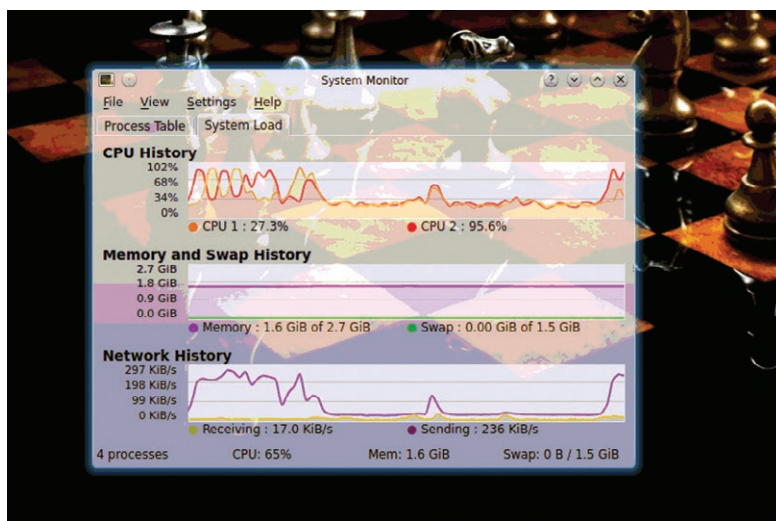
Работать с *Remmina* для доступа к Raspberry Pi, когда скорость соединения была искусственно ограничена до 56К, было малопривно. Однако если как следует настроить глубину цвета и качество, ее, по крайней мере, можно было использовать, если, конечно, вы отличаетесь терпением. Однако в подобных суровых условиях текст крайне трудно читать, а изображения рассматривать.

Хотя установка *TigerVNC* на Raspberry Pi требует использования экспериментального издания Raspbian, Jessie, в целом ситуацию с низкой скоростью можно исправить с помощью расширенного кодирования *TigerVNC*. И хотя нам он показался весьма впечатляющим, все же на скорости как при коммутируемом доступе он определенно не создает ощущения локально рабочего стола, так что чудес не ждите.

У *X2Go* были сложности с установкой соединения на скорости модемного доступа, но он с ними справился, используя сжатие *Tight* и цветовую палитру 4K. На этой скорости рабочий стол был практически бесполезен; в частности, отрисовка окон вызывала солидные задержки. К счастью, у большинства скорость доступа все же выше, и при 25 кб/с дела пошли значительно лучше.

Chromium не работает с *trickle*, но это отнюдь не помешало нам использовать команду *tc* для ограничения скорости прямо в интерфейсе. Изначально наш рабочий стол KDE загружался прекрасно, но общаться с ним было практически невозможно, поскольку слишком много времени проходило между действиями и откликом. Очень часто соединение терялось. И, как и раньше, повышение скорости до 25 кб/с сделало работу вполне приемлемой.

А *NoMachine* опять показал превосходные результаты; ну, может, не превосходные, но явно лучшие, чем у остальных. Навигация по рабочему столу оказалась куда более быстрой и динамичной, и хотя текст становился нечитаемым сразу после значительного перемещения окна, он снова читался после того, как все приходило в норму.



Если вы хотите использовать свой рабочий стол при низкой скорости соединения с Интернетом, придется обходиться 25б цветами или изыскать соединение побыстрее.

Вердикт

- NoMachine NX ★★★★★
- TigerVNC ★★★★★
- Chrome Remote Desktop ★★★★★
- Remmina ★★★★★
- X2Go ★★★★★

» NoMachine побеждает — его нестибсаемость нас впечатлила.

Производительность

Кто здесь самый быстрый?

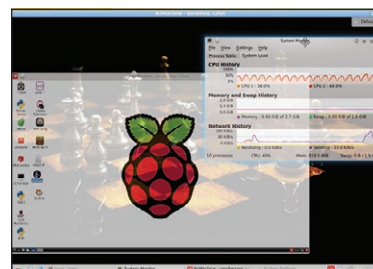
Remmina использует *libvncserver*, где имеется поддержка кодирования *Tight* на *VNC*. Но, возможно, причина была в нашем оборудовании, потому что при настройках *Maximum Quality* соединение через LAN приводило к заиканию нашего рабочего стола KDE 4, в основном при перетаскивании прозрачных окон. Когда мы немного снизили качество, работа стала ровнее, однако трудно избежать артефактов экстрасжатия (особенно в тексте). От применения 16-битного цвета полегчало, но опять же, острый глаз различит некую размытость.

TigerVNC здесь работает немного лучше, будучи в состоянии даже воспроизводить видео с YouTube в полноэкранный режиме. Подобные подвиги возможны только при наличии мощного CPU, в противном случае теряется много кадров.

С этой оговоркой потоковое видео с качеством 720p редко превосходило 5 Мб/с, так что вы вполне сможете делать это на быстром ADSL2+ или кабельном соединении.

X2Go не поддерживает *GLX*, так что у нашего рабочего стола KDE не было эффектов прозрачности, чтобы их можно было с чем-то сравнить. Но даже так он имел намного худший отклик, чем остальные, кроме приложения *Chrome*. Вероятно, причина в перегрузке CPU, которому приходилось обрабатывать все программными средствами. Мы не рекомендуем его для работы, связанной с графикой.

Работать с *Chrome* было возможно, но малопривно. Даже при быстром интернет-соединении он тормозит. Кроме того, он не работает с *Qt5*, впрочем, как и *X2Go*, и это значит, что вы не можете



Здесь был *NoMachine*: это Raspbian внутри KDE и внутри LXQt.

использовать его с *LXQt*. Что огорчает, поскольку это именно тот рабочий стол, с которым он должен бы ладить. Сжатие VP8 работает вполне нормально, все выглядит вполне разборчиво.

Клиент *NoMachine* здесь стал чемпионом. Он один поддерживает *OpenGL*, и воспроизведение игр среднего веса по LAN оказывается вполне возможным. *NoMachine* имеется для Windows, так что это альтернатива стримингу Steam In-Home или *Wine* для игр не под Linux.

Вердикт

- NoMachine NX ★★★★★
- TigerVNC ★★★★★
- Remmina ★★★★★
- X2Go ★★★★★
- Chrome Remote Desktop ★★★★★

» NoMachine (виртуально) вытер об соперников ноги.

Клиенты удаленной работы

Вердикт

И вот он, момент, которого вы так долго ждали, если, конечно, не подглядели таблицу победителей. Как это ни скандально, мы присуждаем первое место *NoMachine* — проприетарному ПО; но у нас не было выбора, настолько он превзошел конкурентов.

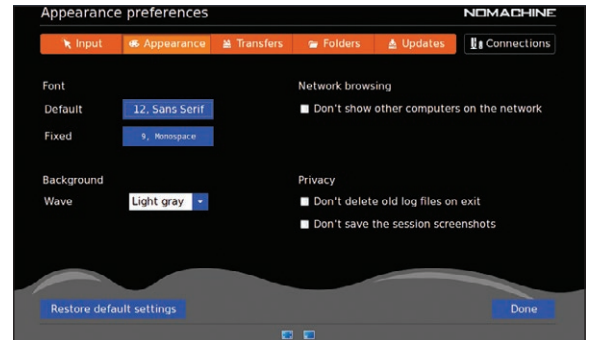
Ситуация способна измениться — в Google работают умные люди, и они могут малость прокачать приложение *Chrome*. Очень хотелось бы увидеть, как прогрессирует *X2Go*, но он потом вполне может занять место среди прочих клиентов в некрополе NX, и это был бы бесславный конец для нашего нового друга Фоки.

NoMachine больше не поддерживает старую версию протокола NX, и дистрибутивы начали использовать *X2Go*-реализацию *pxproхu*; будем надеяться, что это вызовет прилив интереса к данному пакету. Он уже применяется в *Remmina* и Arch Linux, и есть улучшенный клиент Windows под названием *Pyhoca*. Так что потенциал

имеется, надо всего лишь причесать интерфейс: скажем, упростить, или как минимум создать иерархию для невероятного количества предлагаемых схем сжатия.

Интересно, что клиенты VNC по большей части вообще не смогли конкурировать с *NoMachine*, даже *TigerVNC*, который использует высокооптимизированную библиотеку *libjpeg-turbo*. Возможно, это говорит о закате данного протокола — или о начале новой эры господства NX. И опять же, возможно, произойдет подьем в разработке VNC, который совершенно изменит ситуацию вскоре после выхода этой статьи. (Как было с прошлым Сравнением: только мы сказали, что давно не наблюдалось выхода нового основного релиза *VirtualBox*, как вышла v5).

Конечно, некоторые читатели предпочтут открытый код, и здесь отличным выбором будет *Remmina* — если вам позарез нужен удаленный рабочий



» Опция изменения цвета волны в настройках *NoMachine* нас покорила.

стол, вы, вероятно, смиритесь с небольшой потерей производительности. Лишь изредка вы увидите преимущества в использовании *TigerVNC*, и его не стоит рекомендовать детям, которым нужны красивые значки и структуры, а не мерцающее поле для ввода текста.

«Клиенты VNC по большей части вообще не смогли конкурировать с *NoMachine*.»

I

NoMachine NX ★★★★★

Сайт: <http://nomachine.com> Лицензия: Freeware Версия: 4.5
» Проприетарный чемпион... это что ж делается-то?

IV

X2Go ★★★★★

Сайт: <http://wiki.x2go.org> Лицензия: GPLv2 Версия: 4.0.3.2
» Многообещающий, хотя доработать придется немало.

II

Remmina ★★★★★

Сайт: <http://bit.ly/Remmina> Лицензия: GPL Версия: 1.1.2
» Отличная универсальная программа, однако до чемпиона не дотягивает.

V

Chrome RD ★★★★★

Сайт: <http://bit.ly/1GadugV> Лицензия: Freeware Версия: 42
» Милая идея, но ей надо созреть, чтобы догнать остальных конкурсантов.

III

TigerVNC ★★★★★

Сайт: <http://tigervnc.org> Лицензия: GPL Версия: 1.4.3
» Взгляд тигра уже не тот [аллюзия на песню Eye of the Tiger, OST Rocky, — прим. пер].

Обратная связь

Вы довольны нашим вердиктом? Вам не кажется, что Фока милее всех? Не скрывайте своего восторга или злости: lxr.letters@futurenet.com

Рассмотрите также...

Если вас интересует потоковое вещание игр, то, вероятно, *Steam In-Home Streaming* все же сможет побить *NoMachine*. Если у вас видеокарта Nvidia на компьютере с Windows, можете также попробовать использовать программу игрового потока *Limelight* (<http://limelightstream.com>) на клиентской машине, даже если эта машина — Raspberry Pi, с аналогичной целью.

Мы едва упомянули *RealVNC*, но именно здесь трудятся разработчики исходного протокола, версия 5 его свободного клиента вышла в 2012 г., а последнее обновление, 5.2, в феврале этого года.

Однако в серии 5.0 проект отмежевался от лицензии открытого кода, и теперь вам надо зарегистрироваться для получения свободного ключа (с опцией 30-дневного пробного периода

для обладания дополнительными функциями). А не довольно ли проприетарных программ на одно Сравнение? И не забывайте об удаленных клиентах, которые поставляются с самими средами рабочего стола, как, например, *Vinagre* в Gnome и *KRDC* в KDE. Это отличные клиенты, но нам в Башнях **LXF** всегда охота попробовать что-то новенькое. **LXF**

Ubuntu 15.04

Маянк Шарма точно знает, когда наступает весна: зацветают сады и выходит новый релиз Ubuntu.



С самого своего появления Ubuntu всё делал по-своему. Он взялся за труднейшую задачу сделать Linux популярной настольной системой, и мало кто усомнится в том, что этот дистрибутив вывел Linux в массы и породил рекордное количество респинов популярной и стабильной платформы.

И до нынешнего дня, несмотря на то, что он 10 лет умудрялся злить гневливых линуксоидов, Ubuntu остается самым простым из главных дистрибутивов Linux для начинающих пользователей, которые решили перейти на операционную систему с открытым кодом. Если вы выбираете ноутбук с предустановленным Linux, то, где бы вы ни находились, велика вероятность, что единственным возможным вариантом будет Ubuntu. И несмотря на рекордный провал его попытки собрать многомиллионную сумму, вы можете найти смартфоны с предустановленным Ubuntu в продаже в Европе.

Ubuntu завел привычку подливать масла в огонь с каждым новым релизом. Вам придется потрудиться, чтобы найти пользователя Ubuntu, довольного всем. Начиная с революционного интерфейса Unity и заканчивая небезопасными для конфиденциальности Scopes и Lenses, Ubuntu немало потрудились, чтобы стать дистрибутивом, ко-

Самый последний релиз Ubuntu, 15.04, или Vivid Vervet [Веселая Верветка, это такая зеленая мартишка], опять оказался на перепутье, на сей раз из-за своих постоянных попыток стереть грань между рабочим столом и мобильной средой. В этой статье мы подробно разберем этот свежий релиз Ubuntu 15.04, поможем вам разобраться с жаргоном (и слухами), и принять изменения, чтобы лучше справиться с повседневной работой.

Ни умудренные опытом ветераны, ни зеленые новички в стороне не останутся. Новичков мы буквально проведем за руку и расскажем, как выделить место для дистрибутива, потеснив свой Windows. Когда Ubuntu установится на вашем компьютере, мы поможем вам индивидуально настроить его. И если основному релизу Ubuntu все же не удастся произвести на вас впечатление, мы познакомим вас с официально поддерживаемыми альтернативами.

«Многие новшества в грядущем Windows 10 уходят корнями в Ubuntu.»

торый многие активные пользователи Linux так любят ненавидеть. И тем не менее дистрибутив каждый раз возвращался на вершину и возвращал своих пользователей, благодаря нескончаемым обновлениям и улучшениям. О качестве дистрибутива свидетельствует тот факт, что многие новшества в грядущем Windows 10 уходят корнями в Ubuntu (и в Linux в целом).



Ubuntu vs Windows

Вот как Ubuntu обходит конкурентов.

Всё еще сомневаетесь в преимуществах Ubuntu перед вашей любимой проприетарной ОС? Позвольте нам навеки развеять ваши сомнения. Вместо того, чтобы воспевать свободу и гибкость программ с открытым кодом (хотя это по-прежнему весьма значимый пункт превосходства), мы поговорим о преимуществах, оценивая те пункты, которые вы, как регулярный пользователь настольного ПК, тоже можете оценить, ибо сталкиваетесь с ними постоянно.

» **Рабочий стол** И Windows с Metro, и Ubuntu с Unity отказались от старых интерфейсов и перешли на революционно новые, которые заставили пользователей заново осваивать навигацию по рабочему столу. Но если изменение в Windows было весьма радикальным и означало крайнее неудобство в использовании практически для всех пользователей, интерфейс Ubuntu был более интуитивным, и благодаря постоянным и быстрым улучшениям сумел впечатлить даже самых страстных противников.

Unity всегда придерживался стиля интерфейса WIMP (Windows, Icons, Menus, Pointers — Окна, Значки, Меню, Указатели). Да, кнопки переехали на другую сторону окна, но нам все же не пришлось изучать новые способы работы с ними. Опять-таки, хотя на рабочем столе Unity нет традиционного меню навигации, его панель задач со значками является лучшей идеей для начала работы, чем идея Windows заменить значки на плитку.

Windows 8/8.1 явно разработан для сенсорных интерфейсов, тогда как Unity намного лучше справляется с работой и на традиционных, и на сенсорных устройствах. Затем, имеется Dash, который позволяет искать приложения, файлы, музыку и видео. Мало того, для некоторых приложений и web-сервисов, таких, как Ask Ubuntu и Google Books, вы можете установить Lenses и осуществлять поиск через них, не покидая своего привычного рабочего стола.



» Если обновление Windows не позволяет вам загрузить Ubuntu, в дистрибутиве Rescatux (www.supergrubdisk.org/rescatux) вы одним щелчком восстановите Grub.

Мы можем бесконечно приводить доказательства того, что интерфейс Metro — это катастрофа с точки зрения удобства работы. Если вы пытались совместить Metro и традиционные приложения, то вам понятно, о чем мы говорим. Однако больше всего в Windows нас огорчает то, что он принуждает пользователей использовать рабочий стол исключительно по прихоти экспертов юзабилити в Редмонде.

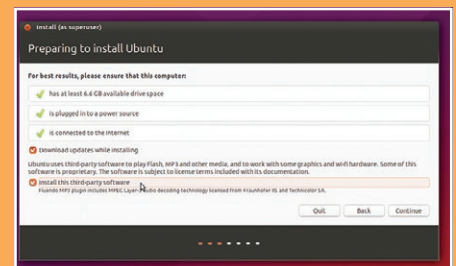
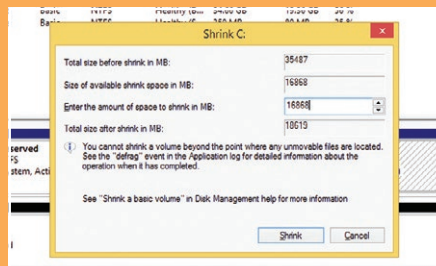
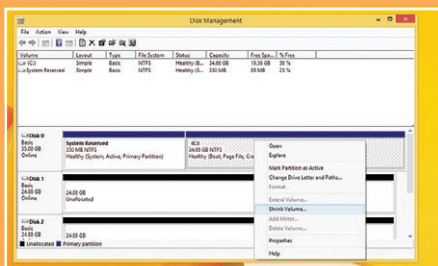
» **Приложения** Чистая установка Windows 8.1 не даст вам того же, что вы получаете от свежей установки Ubuntu. Хотя в Windows 8.1 и больше приложений, чем в предыдущих редакциях, у многих из этих новоиспеченных приложений имеются явные ограничения. Не стоит считать эти приложения заменой настоящим продуктивным программам: пожалуй, лучше всего сравнить их с аналогами на планшете Android.

Сравните его с Ubuntu 15.04, который предлагает отборнейшие программы с открытым кодом, включая web-браузер Firefox, пакет LibreOffice, клиент обмена сообщениями, программы чтения



Установим Ubuntu

Измените размер своих дисков и создайте двойную загрузку Windows с Ubuntu.



1 Выделите место

Чтобы создать пустой раздел для Ubuntu, надо сначала сжать существующий раздел Windows. Запустите инструмент Disk Management в Windows и щелкните правой кнопкой по основному разделу; обычно этот диск именуется C. Затем выберите опцию Shrink Volume [Сжать том] во всплывающем меню.

2 Сожмите Windows

Появится диалоговое окно Shrink и покажет максимальный объем, который вы можете выделить из раздела. Чтобы создать новый раздел, укажите размер раздела в мегабайтах (МБ) и нажмите на Shrink, чтобы начать процесс. Ubuntu нужно не менее 5000 МБ, однако идеально было бы 20000 МБ.

3 Обновления и плагины

Загрузившись со средства установки Ubuntu, ваш ПК отобразит некий перечень. Вы должны отметить два имеющихся окошка на этом экране. Первое будет скачивать все имеющиеся обновления из Интернета, а второе устанавливает плагин для воспроизведения MP3.

электронных книг и медиа-плееры, способные воспроизвести все виды медиа-файлов. Ubuntu уважает условия лицензий разных кодеков, и ради некоего экзотического формата вам, видимо, придется скачать кодек из репозитория, но это потребует всего пары нажатий на кнопку и нимало не затронет ваш кошелек.

Для случаев, когда надо доукомплектовать свою систему, и в Windows, и в Ubuntu есть магазины приложений, хотя пользователи Linux находили и устанавливали программы из онлайн-репозитория с незапамятных времен. Даже пресловутый «новый» *Ubuntu Software Center* создан на три года раньше, чем *Windows Store*. Он позволяет устанавливать важные для системы компоненты и драйверы и даже популярные сторонние приложения, и в нем нет поддельных злонамеренных приложений, которые с завидным постоянством умудряются просачиваться в *Windows Store*.

» **Учетные записи онлайн** Еще одна область, где Ubuntu превосходит *Windows* благодаря своему потрясающему опыту — интеграция онлайн-аккаунтов с приложениями рабочего стола. С помощью функции *Ubuntu Online Accounts* вы можете связать свои учетные записи с популярными сервисами, включая *Facebook*, *Flickr*, *Gmail*, *Google Docs*, *Google+*, *Picasa* и т.д., и получать доступ к своим данным через готовые приложения. От *People App* вы того не дожидаетесь.

» **Игры** Компания *Valve*, решив частично переориентироваться на Linux и выпустив соответствующий дистрибутив *Steam OS* и клиент для своей игровой платформы *Steam*, одной левой поби-

LXF (www.linuxformat.com/forums). Но сейчас давайте сосредоточимся на установке Ubuntu на ваш ПК.

Подготовка к установке

Перед установкой Ubuntu следует познакомиться с некоторыми важными техническими деталями, которые помогут решить потенциальные проблемы, способные возникнуть в будущем. При установке Ubuntu программа установки предложит выбрать файловую систему для раздела Linux. Не вдаваясь в подробности, можно сказать, что файловая система управляет хранением и доступом к данным на вашем диске. *Windows* использует файловую систему NTFS, а Linux — файловую систему ext, самая последняя версия которой — ext4. Ubuntu нужен меньший по размеру раздел подкачки [swap] для пополнения физического ОЗУ на вашем ПК. Неплохой практикой будет создать раздел swap в два раза больше объема вашего ОЗУ, но не более 2 Гб. Если ваш ОЗУ превышает 2 Гб, создайте раздел swap объемом, равным объему ОЗУ.

Еще одно понятие, с которым вам надо познакомиться — загрузчик [bootloader]. Это небольшая программка, которая сообщает компьютеру, где искать операционные системы на диске. В большинстве дистрибутивов Linux используется загрузчик *Grub 2*. Кроме того, жесткие диски в Ubuntu маркируются не буквами, а в соответствии с драйвером устройства, который они используют. Обычно жесткие диски используют драйвер массового хранения [storage], и потому имеют префикс **sd**. За этим префиксом следует буква для указания на конкретное устройство: так, **sda** — это первый жесткий диск. Отдельные разделы определяются цифрой, приложенной к имени жесткого диска. Например, **sda2** — это второй раздел на первом жестком диске.

Дистрибутивы Linux распространяются в виде ISO-образов. Это специально созданные архивные файлы оптического диска, с расширением **.iso**. Вы можете записать их на CD или DVD, в зависимости от их объема, с помощью опции записи образов ISO. Более эффективный и экономичный способ использования ISO-образов — переместить их на диск USB. В *Windows* вы можете применить такие инструменты, как *Unetbootin* (<http://unetbootin.sourceforge.net>) и *Yumi* (<http://bit.ly/LXFyumi>), которые создадут загружаемые USB-диски с ISO-образами многих популярных дистрибутивов, включая Ubuntu. Потом вы сможете употребить эти USB-диски для установки Ubuntu, а затем перформатировать для обычного использования. И, наконец, в *Windows 8/8.1* перед установкой Ubuntu надо отключить *FastStartup*, через *Панель управления*.

Что такое Grub?

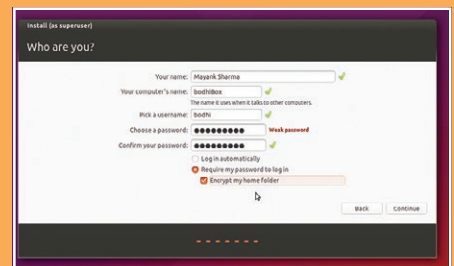
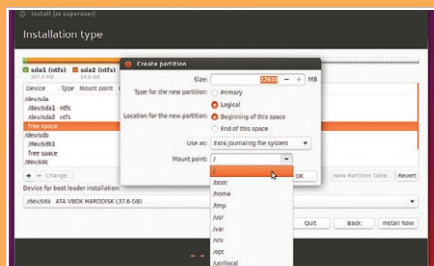
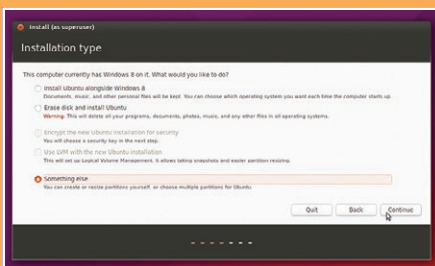
Возможно, вы знаете о загрузчике *Windows*. При переходе на Linux вашему ПК нужно то, что позволит выбрать и запустить нужную ОС. *Grub* — система с открытым кодом, которая делает это в момент загрузки.

«Даже Ubuntu Software Center создан на три года раньше, чем Windows Store.»

ваей самый распространенный аргумент против Ubuntu как основной настольной ОС. При наличии более 1000 игр от *Steam* и *Humble Bundles*, Ubuntu становится интересным и для игроков.

» **Унификация релиза** В отличие от вариаций релизов *Windows* с различными ограничениями функциональности, настольный Ubuntu един для всех. Один и тот же ПК справится с изготовлением презентаций, кодированием, удаленным контролем другого ПК и играми. И все эти задачи вы сможете выполнять одновременно, если способны на подобную многозадачность.

Не составит труда обнаружить еще с десяток других преимуществ. Если вам мало этих доводов, загляните к нам на форумы



» 4 Используйте свободное место

На экране под названием *Installation type* [Тип установки] отметьте переключатель *Something else* [Другое], чтобы вручную разделить диск на разделы. Теперь Ubuntu покажет вам список разделов на диске. Выберите тот, что отмечен как свободное место, и щелкните по значку +, чтобы создать раздел на высвободившемся от *Windows* месте.

5 Определите разделы

В окне *Create partition* [Создать раздел] введите объем раздела Ubuntu, оставив достаточное пространство для раздела swap [подкачка]. Затем в выпадающем меню *Mount point* [Точка монтирования] выберите опцию /. Аналогично создайте раздел для swap, но вместо опции по умолчанию *Ext4* выберите опцию *swap area*.

6 Учетные данные

Ну вот, с этого момента установщик начнет обустраивать Ubuntu. Пока файлы будут копироваться на диск в фоновом режиме, установщик попросит вас указать свое местонахождение и часовой пояс, а также требуемую раскладку клавиатуры. На последнем экране установщик предложит ввести выбранные вами логин и пароль.



Работа в Ubuntu

Как освоить свою новую операционную систему.

Хотя внешне Ubuntu, возможно, и отличается, но работа с этой ОС похожа на работу с другими ОС, поскольку ее интерфейс Unity придерживается традиционного подхода к рабочему столу. Работу Unity по умолчанию можно и переинициализировать, но на самом деле это результат очень глубоких исследований Canonical в области юзабилити.

Слева на рабочем столе Unity располагается Launcher [Запуск] со значками. Первый значок Launcher запускает Dash — это переосмысление традиционных меню в стиле Unity. Вы вводите желаемое в окне поиска Dash вверху и ищете приложения, файлы, музыку, видео, плейджеры, контакты и прочую информацию. Помимо доступа к приложениям и файлам, Dash можно использовать для установки и удаления приложений и предварительного просмотра медиа-файлов. Кроме того, Unity включает Heads Up Display (HUD), инновационную интерпретацию меню приложений. Использование HUD поможет избежать проблем поиска опций, располагающихся где-то в недрах меню.

Персонализируйте рабочий стол

Вы можете сделать свой Ubuntu индивидуальным с помощью встроенных опций конфигурирования дистрибутива. Все настройки Ubuntu находятся в опции System Settings, упрятанной под значком с гаечным ключом и шестеренкой в Launcher.

Опция Personal предлагает настройки внешнего вида и функционирования вашего рабочего стола. Через раздел Hardware вы можете настраивать имеющиеся периферийные устройства, а в разделе System размещаются настройки для изменения компонентов, касающихся установленной системы в целом, например, резервного копирования.

И хотя для большинства пользователей эти встроенные опции вполне адекватны, для большей степени контроля над структурой и удобством в работе надо будет установить дополнительные инструменты. В целом, Unity является плагином оконного менеджера Compiz. Чтобы детально настроить разные аспекты Compiz, потребуется приложение CompizConfig Settings Manager (CCSM), которое можно установить из Ubuntu Software Center или из терминала, командой

```
sudo apt-get install compizconfig-settings-manager.
```

Тем, кто жаждет модифицировать Unity, следует установить инструмент Unity Tweak Tool, который помогает индивидуально настроить многие опции конфигурации рабочего стола и системы. Он имеется также в официальных репозиториях, и его можно установить с помощью

```
sudo apt-get install unity-tweak-tool
```

Интерфейс этого инструмента напоминает интерфейс Ubuntu System Settings, где несколько настроек сгруппированы по категориям под разными ярлыками. Некоторые настройки располагаются поверх предлагаемых Ubuntu, например, Launcher, но вы также можете использовать его для модификации или отключения некоторых основных функций Unity.

Далее имеется QLE Quick List Editor, который умеет редактировать и создавать правой кнопкой quicklist для значков в Launcher. Чтобы установить его, сначала надо добавить его репозиторий с помощью

```
sudo add-apt-repository ppa:vlijm/ql
```

```
Затем обновите репозиторий —
```

```
sudo apt-get update
```

и, наконец, установите инструмент:

```
sudo apt-get install ql
```

Освоившись в Ubuntu, вы сможете перейти к более индивидуальной настройке и настройке загрузчика Grub 2. Для этого понадобится Grub Customizer. И снова — вам нужно добавить его PPA, через

```
sudo add-apt-repository ppa:danielrichter2007/grubcustomizer
```

затем обновить репозиторий —

```
sudo apt-get update
```

и установить инструмент:

```
sudo apt-get install grub-customizer
```

Для воспроизведения контента во всех мультимедиа-форматах установите пакет дополнений с ограничениями:

```
sudo apt-get install ubuntu-restricted-extras
```

Установится несколько пакетов, которые Ubuntu не может на законных основаниях включить в дистрибутив. Кроме того, данная команда установит Flash-плагин на 32-битные дистрибутивы, а вот пользователям 64-битных систем для воспроизведения Flash придется установить пакет *flashplugin-installer*.

Официальные репозитории Ubuntu также включают версию Java с открытым кодом под названием OpenJDK. Ее можно установить с помощью

```
sudo apt-get install openjdk-7-jre
```

```
sudo apt-get install icedtea-7-plugin
```

Если вы предпочитаете установить версию Java от Oracle, добавьте его репозиторий:

```
sudo apt-add-repository "deb http://www.duinsoft.nl/pkg debs all"
```

затем импортируйте ключ GPG:

```
sudo apt-key adv --keyserver pgp.mit.edu --recv-keys 5CB26B26
```

Обновите репозитории по **sudo apt-get update** и установите пакет:

```
sudo apt-get install update-sun-jre
```

Ubuntu Software Center дает вам доступ к ряду превосходных программ с открытым кодом. Мы поощряем вас дать им шанс, но вы можете использовать большинство своих любимых проприетарных приложений. Если вы, например, используете кросс-платформенный клиент Dropbox, вы можете установить его Linux-версию на Ubuntu с помощью

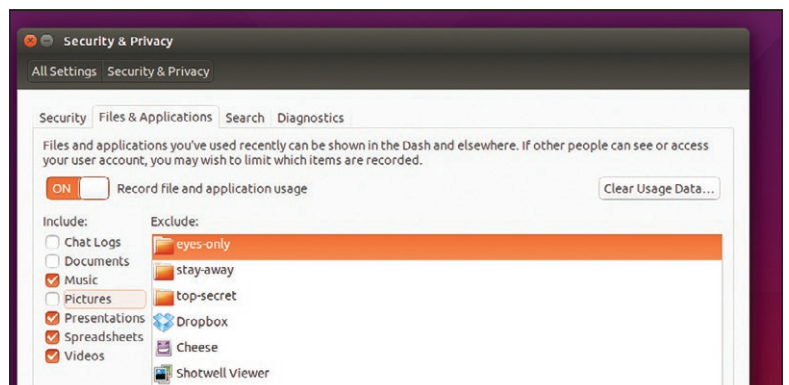
```
sudo apt-get install nautilus-dropbox
```

что скачает все необходимые пакеты и запустит мастер настройки для соединения Ubuntu с вашей учетной записью Dropbox.

Если вы находите музыку через Spotify, вы можете установить его бета-клиент Linux, который работает весьма неплохо. Добавьте репозиторий с помощью:

Терминал

Вы слышали о терминале? В Linux вы будете использовать текстовый ввод команд, который также именуется Интерфейсом командной строки [Command Line Interface] или CLI. Это обычный текстовый интерфейс для запуска команд, однако по большей части вы просто будете вводить заранее определенные строки, которые можно скопировать и вставить через меню.



» Контролируйте свою конфиденциальность с помощью настроек Ubuntu Security & Privacy.

Откуда драйшки?

В целом, драйверы Linux не нужны: они встроены напрямую в ядро. Поэтому большая часть оборудования просто подключается и работает. Это не значит, что в Linux драйверов нет: яркий пример их существования — видеокарты.

```
sudo apt-add-repository "deb http://repository.spotify.com stable non-free"
```

и затем ключ:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recvkeys 94558F59
```

Затем обновите репозитории и установите клиент:

```
sudo apt-get update
```

```
sudo apt-get install spotify-client
```

Некоторые производители программ, такие, как Google, размещают на своих сайтах Linux-версии своих приложений. Например, вы можете скачать Linux-версию браузера Chrome с <https://tools.google.com/chrome> и дважды щелкнуть по скачанному файлу, чтобы его установить. Все пакеты Linux от Google автоматически изменяют соответствующие настройки репозитория, чтобы постоянно обновляться.

Графические драйверы

У большинство современных видеокарт имеются драйверы с открытым кодом. Какой бы ни была ваша видеокарта, Ubuntu позаботится о самом подходящем драйвере с открытым кодом, прямо с момента загрузки дистрибутива. Эти драйверы обеспечивают достойную работу, в том числе при просмотре фильмов и управлении окнами, а иногда и неплохую скорость в простых играх. Но если вы хотите на своем ПК играть, то, в зависимости от вашего оборудования, вам, возможно, придется пройти чуть дальше и найти нужные драйверы.

Установленные в Ubuntu драйверы с открытым кодом работают неплохо, но их основным недостатком является то, что они не обеспечивают полной поддержки всех функций видеокарты и нередко отстают от проприетарных драйверов, поскольку Ubuntu требуется время на включение в пакеты самых свежих драйверов и их размещение в репозиториях. Исключением из этого правила является Intel: их драйверы выпускаются под лицензией открытого кода. (Драйвер с открытым кодом от AMD называется Radeon, а его эквивалент от Nvidia именуется Nouveau.) Далее, следует честно признать тот факт, что некоторые из самых последних игр лучше работают с проприетарным драйвером от производителя видеокарты. Но не напрягайтесь: разнообразные репозитории помогут вам использовать свое видеооборудование в полной мере. Сначала выясните производителя и модель своей видеокарты, командой

```
lspci | grep VGA
```

Если вы используете драйверы с открытым кодом, можете скачать самые свежие драйверы из репозитория Oibaf, они обновляются каждый день. Драйвер добавляется командой

```
sudo add-apt-repository ppa:oibaf/graphics-drivers
```

Если вы используете видеокарту от Nvidia, включите репозиторий X-Swat:

```
sudo add-apt-repository ppa:ubuntu-x-swat/x-updates
```

обновите репозиторий —

```
sudo apt-get update
```

и установите самые свежие драйверы:

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

Если вы воспользуетесь этим репозиторием, не забудьте удалить его перед обновлением релиза Ubuntu. Чтобы удалить внешний репозиторий (или PPA, на языке Ubuntu), установите инструмент *PPA-Purge*:

```
sudo apt-get install ppa-purge
```

и затем удалите PPA X-Swat с помощью

```
sudo ppa-purge ppa:ubuntu-x-swat/x-updates
```

Также, если вы решили перейти на драйвер с открытым кодом от Nvidia, Nouveau, сначала удалите проприетарный драйвер —

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

и затем установите драйвер с открытым кодом:

```
sudo apt-get install nouveau
```

Лучший источник для проприетарного драйвера AMD — сайт производителя (<http://support.amd.com>). Введите информацию о своей карте и скачайте предложенный драйвер. Распакуйте его, и вы увидите скрипт `.run`. Перед установкой драйвера позаботьтесь установить его зависимости с помощью

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

затем запустите скрипт с помощью команды `sh`, например,

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

Запустится *Catalyst*, графическая программа установки проприетарных драйверов AMD, и установит программу управления *Catalyst Control Center GPU*. Встроив самый свежий драйвер видеокарты в свою систему Ubuntu, устанавливайте официальный клиент *Steam for Linux* из *Ubuntu Software Center*. А раз уж вы там, то почему бы не взглянуть на некоторые игры с открытым кодом, например, *O.A.D.*, *Cube 2* и *Battle of Wesnoth*.

Одно из самых обсуждаемых изменений в Ubuntu 15.04 — включение в него *Systemd*, нового менеджера системы и сервисов. Хотя *Systemd* является важной частью ОС, он не слишком влияет на работу обычного пользователя настольного ПК. Однако основные знания и умение работать с этим новым компонентом помогут вам следить за своей системой Ubuntu и поддерживать ее.

Если вы хотите разобраться с *Systemd* поглубже, взгляните в Учебники **LXF184**, стр. 76.

Управление Ubuntu с помощью Systemd

Одной из самых важных задач *Systemd* является управление процессом загрузки, и он хранит подробную информацию об этом процессе. Поэтому если вы хотите знать, сколько времени у компьютера уходит на загрузку, используйте команду:

```
systemd-analyze
```

```
Startup finished in 4.988s (kernel) + 9.674s (userspace) = 14.622s
```

Чтобы выяснить подробности и узнать, сколько времени требуется каждой задаче во время загрузки, введите

```
systemd-analyze blame
```

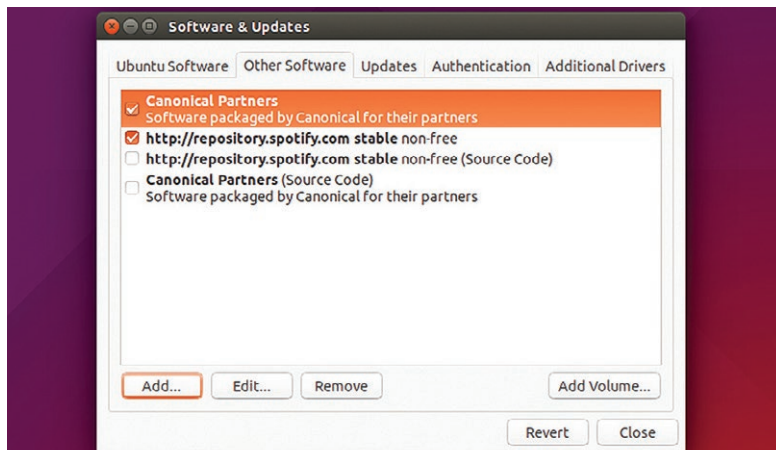
Вы получите список, где указано время, отнимаемое каждым процессом. Для визуализации этой информации введите

```
systemd-analyze plot > boot.svg
```

Вы увидите отображение процесса загрузки и времени, отнимаемого каждым процессом, в файле `boot.svg`. Когда он создается, вы сможете щелкнуть по нему дважды, чтобы просмотреть его при посредстве просмотрщика изображений Ubuntu по умолчанию. *Systemd* умеет также создавать график зависимости установленных сервисов. Для этого вам придется установить программу визуализации *GraphViz* с помощью

```
sudo apt-get install graphviz
```

➤ Из *Ubuntu Software Center* вы сможете установить *Skype*, если включите *Partner Repositories* в инструменте *Software & Updates*.





После ее установки можно создавать графики зависимости целой системы, с помощью

```
systemd-analyze dot --order | dot -Tsvg > systemd-system.svg
```

Можете сузить задачу до составления графика использования вашей рабочей сессии:

```
systemd-analyze dot --user --order | dot -Tsvg > systemduser.Svg
```

или для определенного сервиса, например, для SSH, с помощью

```
systemd-analyze dot 'ssh*' | dot -Tsvg > ssh.svg
```

Чтобы управлять сервисами через *Systemd*, введите команду **systemctl** без аргументов; вы получите длинный список всех сервисов, точек монтирования и устройств с их текущим состоянием и описанием. Для построения более удобочитаемого списка воспользуйтесь командой

```
systemctl list-unit-files --type=service
```

Наличие незагруженных сервисов проверяется командой

```
systemctl --state=failed
```

Вы также можете использовать **systemctl** для получения всяческих подробностей о сервисе. Например, команда

```
systemctl list-dependencies sshd.service
```

предоставит список зависимостей для сервиса SSH. Подобным же образом, заменив **list-dependencies** на **show**, вы получите отображение всех подробностей настройки сервиса. Команда **systemctl** также предлагает некоторые опции контроля сервисов. Команда **systemctl start sshd.service**

запустит SSH-сервер, а **stop** вместо **start** выключит его.

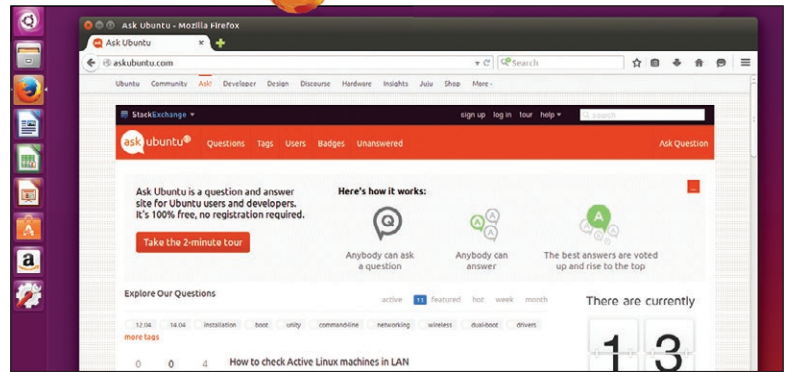
Ни одна из этих команд не остается в силе после перезагрузки. Для включения сервера на постоянной основе, чтобы он загружался автоматически при загрузке компьютера, используйте опцию **enable**, например:

```
systemctl enable sshd.service
```

Подобным же образом, чтобы на постоянной основе отключить сервис, используйте вместо **enable** опцию **disable**. В качестве альтернативы, **restart** остановит и перезапустит сервис, а опция **reload**, если она поддерживается сервисом, будет перезагружать только файл настройки, не прерывая отложенных операций. И, наконец, используйте опцию **status** для выяснения текущего статуса сервиса.

Уровни становятся целями

Предыдущая система инициализации *init* реализовывала концепцию уровней исполнения [Runlevel], то есть предварительно определенные режимы операций, которые определяли состояние машины после загрузки. Они отмечались цифрами. Например, Runlevel 3 переводит компьютер в многопользовательский текстовый режим, а Runlevel 5 также запускает менеджер отображения. *Systemd* заменил концепцию runlevel на цели [targets].



По сути, **target** — это механизм группировки, позволяющий нескольким сервисам запускаться одновременно, что можно использовать для имитации *runlevel*. Но, в отличие от *runlevel*, каждый **target** имеет собственное имя. В *Systemd* есть **target**, которые имитируют общие *runlevel*, такие, как многопользовательский [multi-user] и выключение. Для большей гибкости *Systemd* также разрешает использование **target**, которые наследуют все сервисы от другого **target**, и потом добавляют собственные сервисы.

Список сервисов, связанных с определенным **target**, хранится в директории, которая называется так же, как и **target**, и имеет суффикс **.wants**. Их можно найти в директории **/etc/systemd/system** вместе с символическими ссылками на все сервисы, размещенные в директории **/lib/systemd/systemd**. Чтобы отобразить **target** по умолчанию, используйте

```
systemctl get-default
```

Вы можете переключиться на другой **target** с помощью опции **isolate** — например, чтобы перейти в multi-user **target**, который эквивалентен *runlevel* 3, используйте

```
systemctl isolate multi-user.target
```

Подобным же образом можно переключиться на графический уровень исполнения с помощью

```
systemctl isolate graphical.target
```

Предыдущие команды сразу переключат вас на другой **target**, но чтобы сделать этот переход постоянным, надо применить ключ **set-default**. Например,

```
systemctl set-default multi-user.target
```

на постоянной основе настроит состояние вашего уровня исполнения по умолчанию на неграфический режим.

В *Systemd* содержится куда больше, чем то, о чем мы рассказали в этом кратком руководстве. Однако это всё, что вам надо знать, чтобы управлять своим настольным Ubuntu и анализировать и отлаживать его.

» При проблемах обращайтесь на Ask Ubuntu (www.askubuntu.com) или форумы Ubuntu (www.ubuntuforums.org); вам помогут подсказкой и советом другие пользователи.



»

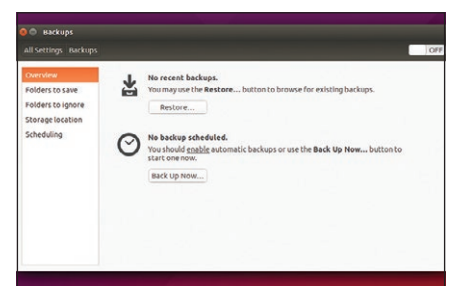
Не забывайте о резервном копировании

Один инструмент, который вам определенно следует настроить, когда вы начнете использовать Ubuntu, это приложение резервного копирования. Если раньше вы не работали с приложениями резервного копирования, то вы будете рады узнать, что Ubuntu предлагает один из простейших инструментов для этой цели, именуемый *Déjà Dup*. Вы можете настроить его, щелкнув по опциям Backups в System Settings.

Инструмент достаточно интуитивен. Осуществляйте навигацию по опциям в панели слева, чтобы определить папки, подлежащие сохранению; место, где вы хотите их сохранить; и график резервного копирования. По умолчанию инструмент настроен

так, чтобы создавать резервную копию всей директории **home**, что, может, и удобно, но обычно это перебор. Вместо этого стоит просто включить такие директории в вашей директории **home**, как **Downloads** и **Documents**.

Приняв решение о том, какие данные копировать, запланируйте место их хранения. Хранить скопированные данные на другом разделе того же самого диска может оказаться довольно бесполезно, если сломается весь жесткий диск. Возможным решением будет хранить резервные копии на отдельном диске или на внешнем носителе. Или, если у вас хорошая скорость Интернета, отправьте резервные копии на удаленную машину.



» Приложение резервного копирования Ubuntu позволяет также определить политику удаления для старых резервных копий.

Разновидности Ubuntu

Что представляют собой официально поддерживаемые производные.

Ubuntu Mate

<https://ubuntu-mate.org>

Одна из самых громких новостей цикла релиза Ubuntu 15.04 — включение дистрибутива Ubuntu Mate в число официально поддерживаемых респинов. Как видно из его названия, этот дистрибутив использует среду рабочего стола Mate, являющуюся ответвлением рабочего стола Gnome 2. У Mate не было трудностей с созданием пользовательской базы из пользователей, разочарованных новыми рабочими столами.

В общем и целом, если вам нравится внешний вид рабочего стола Ubuntu былых времен, то понравится и новая редакция Mate. Помимо того, что он имитирует рабочий стол Gnome 2, Mate придерживается взглядов традиционного рабочего стола, внедряя при этом усовершенствования по удобству в работе из более новых версий, например, привязку окон.

Рабочий стол предлагает знакомую панель сверху, где размещены меню Applications с категориями, меню Places и меню Systems со ссылками на все настройки. Дополнительно дистрибутив предлагает несколько альтернативных раскладок, которые позволят создать среду, похожую на Windows 7 и Mac OS X. Возможно, вы также захотите заменить традиционное меню приложений Gnome 2 на меню Mate, включающее все удобства — строку поиска и быстрый доступ к часто используемым приложениям. В рабочем столе Mate имеется *Control Center*, который позволяет изменять стиль рабочего стола и настраивать оборудование и системные инструменты.

Из-за невысоких системных требований Mate, Ubuntu Mate также позиционирует себя в качестве альтернативы Ubuntu для компьютеров, недостаточно мощных, чтобы работать со сложными рабочими столами. Однако в отличие от специализированных респинов Ubuntu для старых компьютеров, выбор пакетов Ubuntu Mate демонстрирует большее внимание к функциям, нежели к легковесности. Поэтому вы найдете в Ubuntu Mate 15.04 такие мощные и полнофункциональные приложения, как *LibreOffice*, *Firefox*, *Thunderbird*, *Shotwell*, *Pidgin* и *VLC Media Player*.

Дистрибутивы

Дистрибутив — это основанная на Linux ОС, состоящая из набора программ и пакетов, обычно в виде ISO-образа диска. Ubuntu и каждая его разновидность здесь являются дистрибутивами, они могут быть созданы для определенных задач, типов ПК, и т. д. Есть даже дистрибутив Джастина Бибера <содрогаюсь>.

Однако в плане удобства в использовании они различаются. Базовый рабочий стол Gnome 3 имеет ряд особенностей, например, запрет размещать значки на рабочем столе и отсутствие кнопки минимизации на окнах. А если вам не нравится поведение Gnome по умолчанию, можете индивидуально настроить рабочий стол, используя любое из официальных или сторонних расширений. И, как и Ubuntu, Gnome тоже имеет сторонний инструмент индивидуальной настройки, под названием *Gnome Tweak Tool*, который дополняет встроенную панель настроек.

Дистрибутив Ubuntu Gnome 15.04 включает *Gnome Tweak Tool* и *Dconf Editor*, который можно использовать для работы с расширенными настройками Gnome. Дистрибутив также включает приложение *Gnome Online Accounts*, подобное тому, которое имеется в основном дистрибутиве Ubuntu, с рядом приложений, которые пользуются этими онлайн-сервисами. Два из важнейших — *Gnome Documents* и *Gnome Contacts*, которые помогут вам управлять как оффлайн-, так и онлайн-контентом.

Дистрибутив включает такие популярнейшие приложения, как *LibreOffice*, *Evolution*, *Shotwell*, *Rhythmbox* и *Totem*. Поклонники Gnome заметят, что дистрибутив не включил многие приложения Gnome по умолчанию и не основан на самой свежей версии рабочего стола Gnome. Однако у дистрибутива существует привычка выдавать последний релиз Gnome в доступ через официальный PPA.



Kubuntu

<http://www.kubuntu.org>

Основанный на KDE дистрибутив является одним из старейших и популярнейших респинов Ubuntu, а рабочий стол KDE — один из самых отлаженных и качественных рабочих столов. Kubuntu 15.04 основан на KDE и включает интерфейс Plasma 5.2 с такими изящными визуальными эффектами, как новые значки и расширенные темы. У рабочего стола имеется встроенный композитор и множество разных раскладок или видов на языке KDE. KDE знаменит своей настраиваемостью и предлагает больше опций, чем во всех остальных рабочих столах, вместе взятых. Настройки попадают на каждом углу; кроме того, есть центр настроек System Settings, позволяющий индивидуально отрегулировать практически любой аспект самого KDE и даже присоединенного периферийного оборудования.

Две самых выдающихся функций рабочего стола KDE 4 — это виджеты и activities [комнаты]. Практически все на рабочем столе является виджетом, и вы можете размещать их на рабочем столе и на панели инструментов. Activities используются для создания

Помимо «классического» дистрибутива, Ubuntu поддерживает ряд производных.

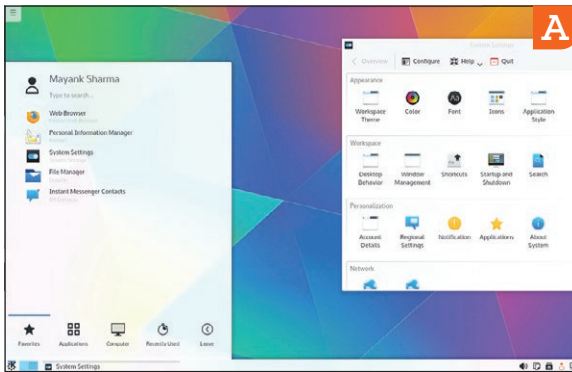
- A KDE
- B Gnome
- C Mate



Ubuntu Gnome

<http://ubuntugnome.org>

Наглядно говоря, рабочий стол Gnome 3 выглядит и работает очень похоже на рабочий стол Unity основной редакции Ubuntu. По сути, рабочий стол Gnome 3 предлагает те же элементы, что и Unity Dash, типа Activities Overview, который содержит подобную Launcher панель Favourites, которую можно индивидуально настроить, прикрепив к ней часто используемые приложения.



рабочего стола четко выраженной направленности, с учетом контента: такова, например, социальная комната [social activity], которая откроет для вас все ваши учетные записи быстрого обмена сообщениями и покажет обновления и ленты разных соцсетей.

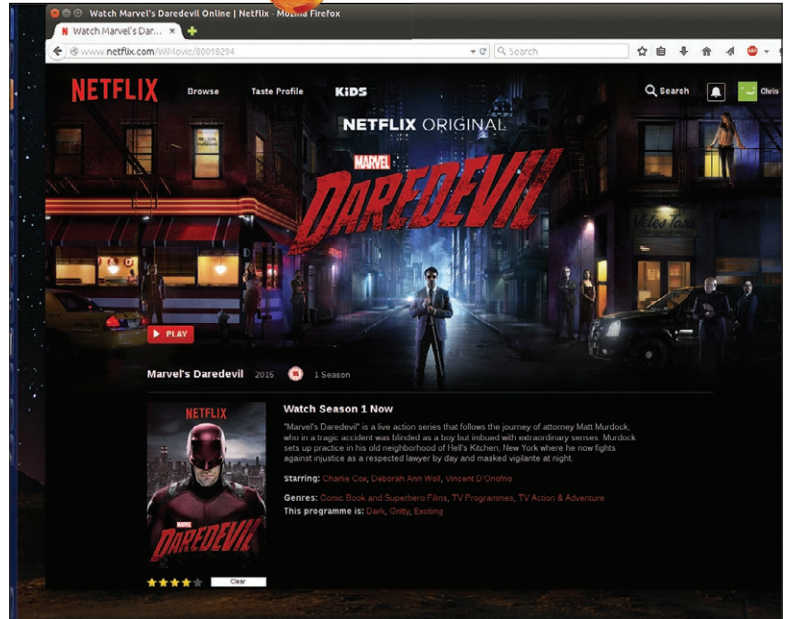
Kubuntu предлагает лучшие приложения KDE, в том числе *Kmail*, *K3b*, *Amarok* и *Dragon Player*, а также популярные сторонние приложения, такие, как *Firefox* и *LibreOffice*. В строке состояния также имеется приложение *Instant Messaging*, которое использует среду KDE *Telepathy* для соединения с десятками сетей IM, в том числе с Google Talk, Facebook, AOL, Windows Live, Yahoo Messenger и т. д.

Специализированные респины

Помимо респinov, ориентированных на рабочий стол, есть четыре других официальных респина Ubuntu, которые сосредоточены на функциях:

» **Xubuntu.org** Этот релиз основан на рабочем столе *Xfce*, который хорошо работает на не слишком старом оборудовании. Он включает *Xfce 4.12*, являющийся самым последним релизом этого рабочего стола, появившимся в результате трехлетней разработки. Дистрибутив также использует программу запуска приложений *Whisker*, где имеются такие современные функции, как строка предпочтений. Неудивительно, что Xubuntu не включает полнофункциональных офисных пакетов, а только текстовый редактор *AbiWord* и редактор электронных таблиц *Gnumeric*.

» **Lubuntu.net** Если вы хотите получить все блага Ubuntu на старой машине, вам нужен дистрибутив Lubuntu 15.04. Он основан на легковесном рабочем столе *LXDE* и использует более



» Вы можете смотреть Netflix в Ubuntu через браузер Chrome.

легкие альтернативные приложения, такие, как *mtPaint*, *Abiword* и *Gnome Mplayer*.

» **UbuntuStudio.org** Помимо всех прочих, есть еще Ubuntu Studio, где предусмотрены инструменты для создания и редактирования мультимедиа. Этот дистрибутив — самый увесистый из всех официальных респinov Ubuntu: он включает практически все инструменты с открытым кодом, в том числе такие тяжеловесы, как *Audacity*, *Rosegarden*, *PiTiVi*, *Kino* и *Blender*. И все эти приложения, как и многое другое, аккуратно организованы в меню под названиями Audio Production, Video Production, Photography, Publishing и т. д.

» **UbuntuKylin.com** Это официальный респин Ubuntu, созданный для китайских пользователей в сотрудничестве с Министерством информационных технологий Китая.

Итак, теперь, когда вы установили, попробовали и освоили *Systemd* и открыли для себя все виды Ubuntu, пора двигаться дальше — и освоить Linux. Впереди вас ждет целая жизнь захватывающих открытий. **LXF**



Следующий Ubuntu

Canonical, разработчики Ubuntu, наконец-то готовы оправдать имя своего рабочего стола, Unity [Единство], реализуя его на разных форм-факторах. По словам разработчиков, это логичная эволюция среды, созданной изначально для настольного ПК и переместившейся на другие устройства — телевизоры и планшеты. Следующая версия рабочего стола, которую окрестили Unity 8, станет первой действительно совмещенной реализацией Unity.

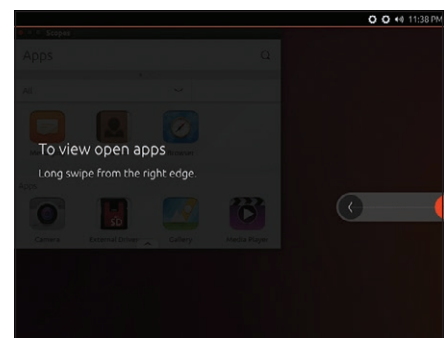
Разработчики, включая самого Марка Шаттлворта [Mark Shuttleworth], уверили пользователей, что работа с Unity 8 не будет сильно отличаться от текущей Unity 7, но можно с уверенностью рассчитывать на заметные визуальные различия. Unity по-прежнему будет оболочкой со всеми своими признаками, включая Launcher, индикаторы и Dash. Однако они будут расширены и стилистически улучшены благодаря проекту Ubuntu Touch. Настоящие изменения в Unity 8 пока остаются тайной.

Его начинка полностью переписана. Например, теперь рабочий стол основан на *Qt5/QML*, в отличие от предыдущей версии, которая была написана на собственном наборе инструментов Ubuntu на основе OpenGL под названием Nux. Разработчики объясняют это изменение многочисленными положительными отзывами о работе с *Qt5/QML* в проекте Ubuntu Touch.

Другое значительное изменение — совершенно новый сервер отображения. Для реализации желаемого взаимодействия с пользователем разработчики Ubuntu будут реализовывать Unity 8 на сервере отображений *Mir* вместо традиционной системы X window.

Unity 8 сейчас находится в активной стадии разработки, и пока не готов для широкого использования. Однако разработчики выложили загружаемый ISO ежедневной сборки Unity 8 (<http://bit.ly/DesktopNextDaily>), который позволит

любому пользователю взять новый рабочий стол для ознакомления, с обычными оговорками, присущими ранней стадии разработки кода.



» Скачайте сборку Ubuntu Desktop Next, чтобы познакомиться с Unity 8 на Ubuntu 15.04.



Мэтью Хэнсон беседует с Пурнимой Венкатакришнан и Седриком Уильямсом из PayPal о том, как крупнейшая система онлайн-платежей перешла на Open Source.

Платим вперед



Кто не слышал о PayPal? До недавнего времени, прозрачностью и расположением к Open Source эта система онлайн-платежей не отличалась, но времена меняются. С назначением Дениз Купер [Denise Cooper] на должность главы Open Source Software в прошлом году, компания перешла с проприетарных стеков web-приложений, взаимодействующих с PayPal, на решение с открытым исходным кодом на базе Node.js и собственный слой Kraken.js. Эти концептуальные изменения значительно сказались на работе всей компании, что дало нам повод встретиться с Пурнимой Венкатакришнан [Poornima Venkatakrishnan], web-разработчиком PayPal, а также Седриком Уильямсом [Cedric

Williams], членом правящей когорты, и порасспросить их об этих переменах настроений в PayPal.

Linux Format: Расскажите немного о ваших ролях в PayPal.

Седрик Уильямс: Я работаю в PayPal с 2011 года. Ранее я занимался онлайн-банкингом для First Data, создавал ПО для управления складскими операциями для розничных партнёров, работал в природоохранных органах и других областях.

Пурнима Венкатакришнан: Я работаю в составе инженерной команды разработчиков web-приложений, непосредственно под началом Джеффа [Харрелла, Jeff Harrel], а до этого много занималась мобильными приложениями. Семь лет я проработала на Palm, потом какое-то время пыталась начать своё дело,



а в ноябре 2012 года устроилась в PayPal. С тех пор я всецело занималась переходом на Node.js, и это чрезвычайно увлекательное занятие.

LXF: Вы упоминали об этом в презентации на OSCON в 2014 году. Помнится, один из ваших разработчиков написал в Twitter, что с переходом на Node.js работать стало куда интереснее.

ПВ: Да, это верно. Интересное было время, столько волнений, столько... препятствий, которые нужно преодолевать, но это было замечательно. Тем более, что всё увенчалось успехом, верно? Это было здорово.

LXF: Мы бы хотели поговорить о переходе PayPal от стека web-приложений, то есть проприетарного формата, к стеку с открытым исходным кодом. Чем это мотивировано?

ПВ: Я думаю, что здесь Седрику будет что сказать, но у меня тоже сложилось своё видение, когда я пришла. Пользовательская сторона всех наших проектов была никуда не годной, и ничего не делалось для её улучшения, создания Puppet своего десятилетия. Потом все осознали, что происходит, почему мы никуда не двигаемся. Пришло понимание того, что нужна более простая, более адаптивная модель взаимодействия с пользователем, основанная на обратной связи от клиентов, а не на наших предположениях о том, что им нужно.

Итак, в процессе, мы пересмотрели имеющиеся у нас стеки; они оказались непригодны, поэтому мы стали искать новый инструмент для создания прототипов. Тут мы и наткнулись на Node.js — и поняли, что именно такое нам и нужно. Тогда мы взяли курс на Node.js; правда, позднее он несколько изменился. Появлялось всё больше и больше надстроек.

LXF: То есть, в основном, это делалось для улучшения общения с пользователями?

ПВ: Да, и также формы представления.

СУ: И ради эффективности. Чтобы разработчики пользовательских интерфейсов могли сказать: «Я создал этот UI и выложил не просто скриншот, а работающее приложение, конечно, внутри нашего сетевого экрана, с фиктивными данными и всё такое, но смотрите, я демонстрирую функциональность, которая войдёт в наши сервисы, их тестовые версии. И я сумел сделать это всего за пару дней, а не две недели». И легко будет сказать: «О, это тоже пойдёт в производство!» Без всяких: «Да, нам нужна система быстрого прототипирования, но потом её следует встроить в старый стек» — это должно стать правилом.

ПВ: Не говоря уж о радости разработчиков!

LXF: Да, их надо почаще радовать!

СУ: Ну, почему бы и нет [смеется].

LXF: То есть с переходом на Node.js вы решили, что самое важное — интерфейс пользователя, и начинать надо с него?

ПВ: И безотлагательно. Нашим проектам необходимо было новое лицо, и как можно скорее. Это была одна из самых веских причин.

СУ: Плюс, Node.js в этом силён. Мы не собираемся отказываться от некоторых частей наших прежних систем, выполняющих невероятно сложные вычисления. Node не для этого. Он управляется событиями и на редкость быстро выполняет мелкие транзакции. Не стоит загружать подобное приложение десятичасовым отчётом — это не его сильная сторона. И мы следим, чтобы Node применялся по назначению, там, где он эффективен. Мы также думаем использовать его на уровне служб. Ведь многие из них устроены однотипно, состоят из множества мелких транзакций. Есть и другие, со сложными вычислениями, для них мы его использовать не будем.

Упраздняя старые системы клиентского уровня, мы также отметили, что хорошо бы поддерживать больше нюансов, и стали использовать другие языки. Это и Go, и Scala [см. стр. 80] — чего мы только ни пробуем для решения конкретных задач; раньше мы такого не делали. Приходится учитывать и уже имеющиеся наработки с Node.js, а значит, отбирать. «Этот инструмент подходит для этой задачи, а этот — нет», и незачем применять один и тот же для всего.

ОБ ИЗМЕНЕНИЯХ

«Трудно переломить представление, что „JavaScript — язык несерьезный“.»

LXF: Как Node.js может улучшить жизнь разработчиков?

ПВ: Ну, во-первых, этот язык используется как в браузере, так и в приложениях... и если раньше для разработчиков это были две отдельных роли, теперь возможности стали шире, и они могут свободно переходить из одной сферы в другую, улучшив своё резюме до полноценного специалиста. Это стало проще уже с JavaScript, и Node.js также этому способствует. То, что на Node легко написать, оформить что-либо, немедленно





запустить и посмотреть, как оно работает — уже большая победа... Так что, да, я считаю его одним из главных достижений; с ним так просто начать что-то новое, он подходит как для браузера, так и для всего, что связано с уровнем представления. Естественно, что разработчики клиентских приложений быстро приновились к Node.js.

LXF: Учитывая также время компиляции Node, верно?

ПВ: Да... Принцип работы Node в том, что создаются небольшие модули, каждый из которых автономен и имеет свою задачу. Поэтому сами они должны быть как можно меньше, а покрытие кода как можно больше, чтобы вы свой модуль знали досконально, когда его публикуете. Именно благодаря этому в сообществе Node процессы компиляции и агрегации идут гораздо быстрее.

LXF: Какие сложности возникали в связи с отказом от прежних систем?

ПВ: [Седрику, со смехом] Может, ты расскажешь? Я присоединилась, когда процесс перехода уже начался. Мне повезло, что я этого не видела и могла сразу включиться в работу!

КАК РАЗВЕЯТЬ СКЕПСИС НАСЧЁТ JAVASCRIPT

«Говоришь людям: „кода всего 20%, а задача выполняется в пять раз быстрее“»

СУ: А меня как раз для этого наняли. Это и был мой стимул работать в PayPal. Ведь перед PayPal стояла особая задача. Мы знали, что оказались в изоляции. Мы понимали, что сами загнали себя в угол. И надо было это исправить. И я решил подойти к этому с разных сторон. Сложности возникают с самим внедрением новой технологии. Во-первых, сроки. В PayPal люди очень заняты. У них свой плотный график, и тут появляется вы и говорите: «Посидите пока в сторонке, мы тут на новый стек переходим», а они говорят: «Но надо известить об этом клиентов»... и ты ищешь способ, как добраться до этого приложения... от которого столько зависит... которое нужно как следует протестировать... и вернуть к работе так, чтобы при этом не помешать другим внедрять новые функции и сделать его удобнее для клиентов. Пришлось

потрудиться. И кроме того, внедряется новая технология, которой никто еще не пользовался, нужно находить с операционистами общий язык. И разработчикам приходится впитывать много нового. Трудно бывает переломить представление, что «JavaScript — язык несерьезный». И ещё труднее решить, кто будет это всё поддерживать? И вот есть сервер с работающим приложением, он довольно надёжен. Но сколько таких серверов нам может понадобиться?

ПВ: ...чтобы они ещё вписались в экосистему PayPal, ведь там было много проприетарных вещей, а в Open Source всё по-другому. Поэтому когда я устраивалась в PayPal и узнала, что там работает Дуг Крокфорд [Doug Crockford, предприниматель, создатель JSON и главный разработчик JavaScript в PayPal], я сказала: «ОК, если вы хотите меня нанять, я согласна». Только потому, что там был Дуг.

LXF: Возвращаясь к «JavaScript — язык несерьезный»: что, у некоторых разработчиков настолько твёрдое предубеждение?

СУ: На самом деле, это предубеждение было для нас весьма полезным пугалом. Как только кто-то такое произносит, спрашиваешь: «Правда? А как насчёт функционального программирования на Java? ОК, приведите мне пример функции, которую можно добавить в обратный вызов так, чтобы она завершилась сама, и не пришлось отслеживать её состояние через всю цепочку вызовов». Говоришь людям: «кода всего 20%, а задача выполняется в пять раз быстрее», и они отвечают: «Ух ты! Я об этом не думал», так что я люблю, когда приводят этот аргумент, потому что я знаю, что им важно, и могу рассказать им что-то о JavaScript, и показать: «вот это скорость, вот это мощь, и если вы рассуждаете как функциональный программист, вы добьётесь успеха, повысите свой уровень». И получаете ту же производительность, используя сервер всего на 10%.

ПВ: Можно посмотреть и с философской стороны. Каждый язык обладает своими плюсами и минусами, и надо понять, насколько каждый из них вписывается в ваш стек, чтобы найти и освоить их сильные стороны.

Если JavaScript такой несерьезный, то почему он до сих пор существует? Почему не умер? У меня нет точной статистики, но я знаю, что на GitHub проекты JavaScript очень популярны. Они входят в число лидеров. Всё самое передовое в Open Source сегодня связано либо с клиентскими приложениями, либо с Node.js. Почему тогда «несерьезный»? Разберитесь с этим, а потом уж и говорите.

СУ: И при этом мы можем сказать: «У JavaScript есть и слабые стороны. Не используйте их». Что мы и делаем. У нас есть Дуг Крокфорд, который обучает сотни человек JavaScript, и он спокойно говорит: «Не делайте того-то». Потому что здесь любой язык будет плох.

ПВ: О да, нам повезло учиться у самого Дуга Крокфорда.

LXF: То есть если при выборе языка знать его плюсы и минусы, избегать вторых и опираться на первые, не будет причин целиком отвергать никакой язык?

СМ: Ну, вот Дуг входит в комитет ECMA, занимается созданием ECMAScript 6. И у нас есть Эрик Тот [Eric Toth], один из главных разработчиков Kraken, и тоже член этого комитета, так что мы, можно сказать, не только «учитываем слабости JavaScript», но также и способствуем тому, чтобы в новой версии таковых было поменьше. Надеемся, в 7 будет меньше, чем в 6. Но мы — за JavaScript, мы обязательно будем работать с ним.

ПВ: Но ещё раз, мы понимаем, что он не универсален, и настаиваем, что перед его использованием необходимо проверить, насколько он подходит.

LXF: В чём состоят плюсы перехода на Open Source?

ПВ: У меня своё мнение на этот счёт [*смеётся*]. Делая открытый проект, чувствуешь, что всё — обсуждение, и участие, и владение... и поддержка — осуществляется на равных правах. Вопрос только в том, насколько ты готов в этом участвовать. Вот и всё. Этим всё определяется. Так что если говорить о поощрениях, я не знаю, может быть, исходя из моего прошлого опыта, я считаю, что это прекрасная возможность для тех, кто хочет просто показать, на что он способен, без каких-то ещё стимулов. В этом и прелесть открытого кода: мало того, что ты пользуешься тем, что делают другие, ты и делишься тем, что сделал сам, отдаёшь миру лучшее — лучшее и получаешь. Это такая жизненная установка. И мне она нравится. Мы совсем недавно с Седриком говорили, что в открытом коде есть нечто кармическое. Ведь всё, что ты делаешь — ты делаешь на благо всего сообщества, а если нет — то твой вклад просто исчезнет.

LXF: Нашим читателям очень близка идея сообщества, готовности делиться с другими, безвозмездно, по собственному желанию.

ПВ: Я видела много таких примеров, с тех пор как присоединилась к Node.js. Как-то я была на одном обучающем мероприятии по Node в NPM [менеджер управления пакетами в JavaScript], и постоянно слышала там одну и ту же фразу: «Nice people matter». [По смыслу — «Главное, чтобы человек хороший попался», — *прим. пер.*] Это и есть NPM — Nice People Matter. И мне показалось, что в том и суть идеи открытого кода. И — да, это очень приятно: чувствовать, что ты не просто пишешь код, а делаешь что-то для общества.

СУ: Мне всегда казалось, что эта идея близка по духу сообществу разработчиков PayPal, просто это не было частью официальной политики. Одна из главных задач PayPal — управление рисками: безопасность, надёжность имеют для нас ключевое значение. Многие наши решения определяются желанием сделать всё максимально безопасным. Отсюда и стремление обособиться, и то, что наши сотрудники, желая внести свой вклад в открытый код, не решались стать членами этого сообщества, ведь такой подход к безопасности делает организации более закрытыми. Мы защищаем финансовые данные своих клиентов. И делаем это очень тщательно. Прямо сейчас, превращая PayPal в более открытую структуру, с помощью Дениз [Купер] и других, мы стараемся донести до людей, что доступность возможна без ущерба безопасности, и это более тонкий подход к разработке, определяющий наши решения. И я считаю, нам ещё много над чем предстоит поработать, не только над интеграцией с сообществом и умением делиться нашими потрясающими инженерными достижениями, но и в плане лучшего понимания того, что в тех областях, где безопасность действительно превыше всего, есть другие специалисты, знающие эти факторы риска и угроз лучше, чем те, которые работают в PayPal.

LXF: Для PayPal важно поддерживать уникальный баланс открытости, ведь многим нужна прозрачность системы, чтобы проникнуться к ней доверием, но при этом вы имеете дело с деньгами клиентов, а значит, должны заботиться о безопасности.

СУ: Как мы знаем, самые эффективные меры безопасности — именно открытые. Протоколы шифрования, схемы взаимодействия, аутентификация (сквозная или на основе маркера) — всё это лучше работает тогда, когда о них знают все.

Но есть и такие элементы системы безопасности, о которых, по юридическим или лицензионным причинам, говорить в открытую мы не можем. В управлении рисками есть масса того, что никак не связано с web-сервисами. Мы должны чётко отделять то, что принадлежит только нам, и готовы делимся всем остальным. Сейчас мы этим серьёзно занимаемся.

LXF: А как сообщество Open Source реагирует на ваши обращения?

СУ: Очень прямо, за что им спасибо. Когда мы только выложили первые показатели производительности Node в сравнении: «Вот наш новый стек Node, а вот наш старый проприетарный стек Java», люди быстренько пробежались по цифрам и заявили: «Это некорректное сравнение!» И мы согласились: «Да, спасибо. Вы правы! Некорректное!» И нам пришлось более чётко пояснить свою исходную позицию. Нам важен любой отклик на наши продукты.

ПВ: Продолжу в более позитивном ключе, по крайней мере, в отношении нашего открытого проекта — Kraken. Сообщество дало нам хороший толчок, извещая о проблемах и даже предлагая свои реализации каких-нибудь крупных функций. И это здорово. Мы получаем не только жалобы, но и похвалы, и даже го-

БЕЗОПАСНОСТЬ И OPEN SOURCE

«Самые эффективные меры безопасности — именно открытые.»

товые предложения, как сделать лучше. Это круто. На самом деле, Kraken [в ранней версии] и есть итог обратной связи с сообществом, так что мы решили попробовать ещё раз и попытаться перестроить, переписать его с учётом тех прекрасных идей, которые к нам поступают.

LXF: Расскажите, пожалуйста, как устроен Kraken.

ПВ: Kraken — тонкий уровень загрузки над собственно express-пересылкой. Фактически он определяет структуру и соглашения для ваших приложений. Он максимально настраиваемый: прописав конфигурации, вы можете настроить приложения для конкретной среды, не добавляя никакого кода. Можно просто абстрагировать их как настройки JSON. Абстракция Kraken автоматически настроит их правильно при загрузке. Плюс, когда речь идёт о web-приложениях, важна маршрутизация и, следовательно, промежуточное звено, через которое она осуществляется. »





В Kraken есть удобное правило, позволяющее легко это настроить. В Kraken 1.0 также много улучшений, позволяющих определить порядок обращения к промежуточному звену. Много интересных дополнений — например, таких как интернационализация, для перевода шаблонов. Недавно появилось ещё одно, под названием Специализация шаблонов [Template Specialisation], оно помогает легко провести сквозной тест ваших представлений. На тот случай, если часть из них должна быть помечена по-другому, скажем, для совместного тестирования приложений, либо для разных стран — в PayPal такое часто требуется. Нечто подобное было в любой компании, выбравшей Kraken. Поэтому мы и решили сделать весь код доступным. Так что если сам Kraken — просто уровень архитектуры, то Kraken.js — это целый инструментарий, со множеством дополнительных компонентов, которые пригодятся для любого приложения.

О РАЗВИТИИ OPEN SOURCE

«Мы стараемся, чтобы во всей разработке преобладали принципы открытости.»

LXF: Можете ли вы привести примеры приложений, созданных с Kraken?

ПВ: В GitHub их сейчас уже несколько. Для специализации, для совершения онлайн-покупок через корзину PayPal. Для использования DB Wisard. Всего их порядка четырёх-пяти. Но, помнится, были и другие, на базе предыдущей версии, например, как с Kraken можно использовать сокеты. Это было интересно, поскольку варианты были самые разные, и люди обменивались идеями. С Kraken 1.0 пока этого нет, но такое уже было.

LXF: И много ли предприятий уже используют эти приложения?

ПВ: Мы уже провели переговоры с рядом компаний, я знаю, что Джефф [Харрелл] общался с Ancestry.com. Вместе с ним мы были в Coupons.com... Всего я не упомяну, но были и другие. Так что

такие компании есть, они заинтересованы, хотя быть в курсе, те же Ancestry и Coupons, вероятно, уже пробуют их как-то использовать, потому что в Coupons.com нам уже задавали конкретные вопросы относительно кода... значит, к ним серьёзно присматриваются.

LXF: То есть он взаимодействует не только с PayPal?

ПВ: Нет-нет-нет. Сам Kraken никак не связан с PayPal. Мы не показываем, как его можно интегрировать с API, хотя это отдельный пример. Скорее это абстракция, которую каждый может использовать для создания своих приложений. Это как express, но только в Kraken больше возможностей именно для предприятий.

СУ: В самом PayPal мы рассматриваем Kraken как некую систему координат, дорожных указателей, говорящих: «вот это покроет большинство ваших нужд, сначала попробуйте это». Это не значит, что вы не сможете выйти за их пределы ради того, что важно именно для вашего приложения, но прежде чем сделать это, секунду подумайте. Возможно, для этого есть стандартное решение, понятное всем тем, кто используют приложения Kraken. Именно согласование правил делает их эффективными.

Правило появляется потому, что сообществу оно показалось интересным и полезным, но это не значит, что нет других способов создать приложение, других правил, которые подойдут другим компаниям, и мы надеемся, что они тоже поделятся этим, чтобы мы могли посмотреть и сказать: «О, нам такое тоже пригодилось бы!»

ПВ: Возможно, это уже случилось. Kraken уже много перенял от других фреймворков. Даже внутри PayPal, где родился Kraken, по его моделям легко создавать абстракции для приложений Node в унаследованных частях системы. Так что прямо сейчас, благодаря Kraken, можно вставить связку от PayPal к приложению Kraken, и получится спецприложение PayPal. Проще просто. Без какой-либо подстройки к внутренним системам PayPal.

LXF: И, как вы упомянули, без написания кода?

СУ: Для интеграции с PayPal добавляются наши внутренние модули. Их нужно добавить в конфигурацию, но у нас есть свой собственный внутренний реестр NPM, это каппа-модуль, который выложен на наш открытый сайт GitHub, он позволяет добавлять внутренние модули из внутреннего NPM, а остальное — из внешнего, через проху-запрос. Итак, модули PayPal добавлены в конфигурацию, NPM установлен, и всё работает. Любое приложение теперь сможет работать с внутренними службами. Мне кажется, в любой крупной организации это бы делалось так же. Вот наши внутренние модули, вот проху, вот внутренний репозиторий. Всё остальное — из открытого NPM.

LXF: Какова будет следующая задача открытого PayPal?

СУ: Сложно сказать. В Open Source занято столько людей на разных уровнях... Мы меняем подход ко всей внутренней структуре — стараемся, чтобы во всей разработке преобладали принципы открытости. Даже там, где мы никогда не откроем доступ извне, мы собираемся сделать это для внутреннего пользования, чтобы в создании ПО мог поучаствовать любой сотрудник. Что из всего этого получится — не знаю, но жду чего-то впечатляющего и удивительного. Я рад, что делаю это. Рад бы посмотреть на себя через пару лет и увидеть, что из этого выйдет.

LXF: Похоже, переход части PayPal на Open Source изменил концепцию компании?

ПВ: Это так. Начинаются медленные изменения внутри, что появилось в совместной работе над проектами, обмене идеями, их совместном использовании. И такого взаимодействия стало больше, поскольку появилось больше каналов. Сотрудничество развивается, и по-моему, это здорово. **LXF**

Новое поколение средств защиты

Межсетевые экраны ССПТ, не имеющие IP-адреса

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

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

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

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

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

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

Дополнительное устройство защиты для:

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

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

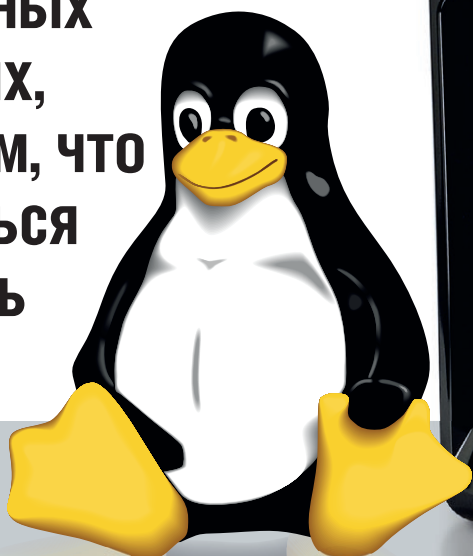
Linux-эксперт для вашего бизнеса. www.linuxcenter.ru

Linux  center

Песнь

об угрозах и борьбе с ними

Джонни Бидвелл проливает свет на мрачный мир вредоносных программ Linux, показывая вам, что может случиться и как избежать риска.



Иногда в пабе можно услышать, как кто-то похваляется, что Linux «безопаснее» Windows. С одной стороны, он прав, потому что пользователям настольного Linux даже близко не надо в той же мере бояться вирусов и вредоносных программ, что их собратьям в Windows. Дело не в том, что вирусов нет: просто они составляют настолько незначительную часть экосистемы вредоносных программ, что становится вполне разумным (при наличии привычки просмотра всего в безопасном режиме) просто не брать их в голову.

Все это сводится к простым цифрам: любой обзор отводит Linux менее 2% от общей доли настольных ПК на рынке. Учитывая все это, авторам

вредоносных программ вполне логично нацелиться на системы Windows и (все больше) Mac. Жертвы могут быть заражены разными способами, обычно через открытие сомнительной ссылки или вложения в сообщении электронной почты

«Пользуясь слабостями Flash, апплеты-вредители могут исполнить свой код.»

либо посещение сомнительного сайта. Иногда используется уязвимость системы, позволяющая злоумышленнику удаленно запустить код на машине жертвы. Взломанные — а иногда и сами вредоносные — сайты могут попытаться потихоньку установить программу-зловреда на компьютер

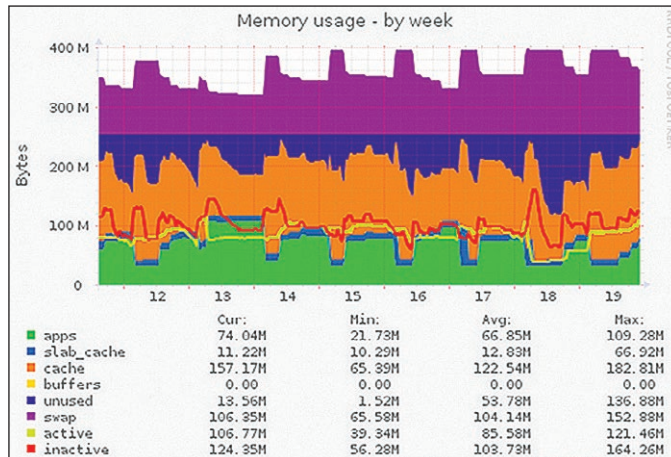
посетителя; такие техники есть. Однако самой распространенной целью атак является плагин Flash. Пользуясь слабостями Flash, апплеты-вредители могут исполнять на удаленной машине свой произвольный код, абсолютно без ведома ее владельца. Очень легко (а порой и оправданно) обвинить Adobe в том, что они загружают сомнительный код; но, опять же, реальная причина в том, что так много людей устанавливают Flash, что злоумышленники просто не в силах этим не воспользоваться. То же самое можно сказать и об Adobe PDF Reader, и о плагине Oracle Java. Chrome 42 отключил официальную поддержку всех плагинов NPAPI, мотивируя это обширной поверхностью атаки, которую они создают для браузера.

Однако сомнительные web-серверы, которые проталкивают вредоносный код на удаленные машины, чаще всего сами являются компьютерами Linux. А тот тип в пабе говорил, что они безопасны... На самом деле, есть бесчисленное количество способов, которыми компьютером Linux в конечном итоге может «завладеть» злоумышленник. А если на нем размещен популярный web-сайт или важная база данных, то это только повышает его привлекательность для подобных злодеев. Наши читатели часто спрашивают нас, как настроить безопасный LAMP или нечто ему подобное, но, к сожалению, простого ответа на этот вопрос нет. Есть ряд вещей, которые вам стоит (и множество тех, которые не стоит) делать, однако невозможно просчитать талантливого злоумышленника или скрытую ошибку нулевого дня в одном из множества компонентов, которые использует современный сервер LAMP. В общем, поговорим о том, от чего нам защититься можно.

Вся суть в настройке

Хотя нарушение системы безопасности может иметь место и в результате некой новой уязвимости с броским названием и стильным логотипом, все же самой основной причиной остается старая добрая неправильная настройка сервера. Сервер с неумеренным количеством разрешений доступа и внесения изменений извне — это сервер, который ищет приключений на свою, э-э, голову. Среди наиболее общих ошибок — разрешение web-серверу вносить записи в важные файлы или доступность SQL-сервера для общественности (хотя ему надо только слушать локально или принимать соединения от избранных IP). Конечно, злоумышленники могут прибегнуть ко взлому логинов SSH (или иных). На самом деле это должно быть невозможно, поскольку пароли для входа обязаны быть отключены (по крайней мере, для особо важных учетных записей), и вместо них должна быть аутентификация с помощью публичного ключа, а неоднократные попытки использовать неверный логин (что в любом случае весьма затратно по времени) должны приводить к временной блокировке.

Итак, проверяйте свои разрешения, заставьте сервер слушать адрес только localhost там, где это возможно (и соединяйтесь через канал SSH, если вам требуется доступ к нему), и настройте



► *Munin* по-норвежски значит «память». Собрав некоторую статистику, станет намного проще выявлять внезапно возросшую потребность в ресурсах.

эффективные правила брандмауэра. В последнем случае будет весьма благоразумно заблокировать не только входящий, но и исходящий трафик. Это может не дать злонамеренно установленной программе связаться со своим источником (они часто общаются по IRC) и учинить разгром. Следует отключить логины root, а авторизованные пользователи должны использовать для выполнения административных задач **sudo** или **su**, поскольку тогда их можно будет отследить через системный журнал. Что ж, предположим, что теперь наш парадный вход вполне надежен и безопасен, но как еще могут злодеи проникнуть на наш компьютер? Это зависит от того, насколько надежно в нем всё остальное.

Типовая область атаки на web-серверы — скрипты PHP, хотя жертвой подобных атак может стать практически любой язык сервера. Везде, где ваше приложение принимает ввод от пользователя, вы не должны терять бдительности. Поскольку вы не можете контролировать вводимую пользователем информацию, очень важно ее дезинфицировать. В противном случае злоумышленник может ввести код, способный в определенном контексте стать вредоносным. Например, простая форма поиска PHP может выглядеть так:

```
<form method='get' action='search.php'>
<input name="search" value="<?php echo $_GET['search'];?>" />
<input type=submit name='dosearch'
value='Search' /></form>
```

Ввод передается непроверенным в скрипт **search.php**, а значит, пользователь может ввести

JavaScript — например, такой, который ищет строковую переменную:

```
<><script>alert(0)</script>
```

В результате появится окно предупреждения. Двойные кавычки в начале останавливают HTML-атрибут **value**, затем правая скобка выходит из элемента **input**. Чтобы защититься от подобных хитростей, обязательно надо применять имеющиеся функции для фильтрации вводимой информации. Следующий код исключит любые спецсимволы, чтобы они не нанесли вреда:

```
<?php
$input = "pointy brackets <and> & ampersands?";
var_dump(filter_var($url,FILTER_SANITIZE_SPECIAL_CHARS));
?>
```

Выводимый в браузере результат будет тем же самым, но если вы посмотрите на исходник HTML, сгенерированный скриптом, вы увидите, что на самом деле он вывел строку

```
"pointy brackets &lt;&gt; & amp ampersands?"
```

Исключенные символы куда менее полезны для злоумышленника. Здесь вы также можете использовать **FILTER_SANITIZE_STRING**, который удаляет (а не исключает) тэги. С таким же успехом вы могли ввести сюда PHP, или — там, где вводимая информация передается в базу данных — команды SQL.

При использовании PHP для взаимодействия с базами данных стоит использовать API PDO (PHP Data Objects) вместо MySQLi. Благодаря этому данные никогда не будут ошибочно приниматься за команды.

Как внедрять обновления, когда их нет

К сожалению, дистрибутивам случается работать на множестве серверов, чей срок поддержки давно истек. Администраторам таких машин стоит объединить свои усилия, но если обновление ОС даже не обсуждается, стоит попытаться бэкпортировать важные отладки безопасности. Иногда некоторые люди щедро предлагают пакеты для устаревающих дистрибутивов, что весьма удобно... но возникает вопрос доверия. В общем, вам придется собирать собственные пакеты, в которые будут входить все

новые отладки системы безопасности. Исходные пакеты для старых дистрибутивов найти легко (для старых версий Ubuntu загляните на <https://launchpad.net>, а Debian — <http://archive.debian.org>).

Очень неплохая идея — настроить виртуальную машину, которая будет как можно более точной копией вашего устаревшего сервера. Вам также понадобится цепочка инструментов *gcc*, настройка которой может включать ад зависимости, плюс зависимости для компиляции пакетов. Вам не надо

будет делать обновлений основных версий уязвимых программ, поскольку это, с большой долей вероятности, поломает вашу систему: вместо этого подстройте новые правки к старой версии, что вовлечет немалую долю проб и ошибок. Если вы используете дистрибутив на Debian, добавьте заплатку в **debian/patches/all directory**, внутри директории исходника пакетов, и добавьте имя заплатки в файл **debian/patches/series**. Затем запустите **debuild** для создания пакета

Будучи обнаруженными и подтвержденными, уязвимости фиксируются в системе Common Vulnerabilities and Exposures (CVE), хотя у отдельных продуктов и компаний могут быть собственные внутренние системы. Если информация о новой уязвимости находится под эмбарго, чтобы не делать ее достоянием широкой общественности, идентификатор CVE все равно можно зарезервировать до тех пор, пока не станет безопасно предавать подробности огласке. Сначала информацию откроют соответствующему кругу людей, чтобы были доступны объявленные заплатки или хотя бы временные решения. Разные дистрибутивы предлагают собственные предупреждения по системе безопасности: например, CVE <https://security.gentoo.org> предлагает централизованный коммутатор для быстрого сбора и распространения информации о срочных и прошлых проблемах.

Не применять заплатки и обновления безопасности означает нарваться на неприятности. Для сравнения, лишь несколько атак являются результатом нападений нулевого дня, а широко распространенные инструменты позволяют злоумышленникам сканировать потенциальные метки на предмет известных уязвимостей. Основные дистрибутивы очень быстро выпускают заплатки для вновь обнаруженных брешей, и очень важно обновлять затронутые этим пакеты, даже если это будет означать незначительные перебои в работе на время перезапуска сервисов. Пять минут простоя и несколько недовольных пользователей — это лучше, чем похищенные данные или необходимость перекомпилировать всю систему, поскольку кто-то в нее проник и установил руткит. Отчет HP Cyber Risk (вышедший ранее в этом году)

сообщает, что 44% брешей в системе безопасности были результатом уязвимостей, известных широкой общественности в течение двух–четырёх лет, и это вина системных администраторов. И еще более грустные статистические данные от Verizon's Data Breach Investigations сообщают, что почти 97% успешных атак были результатом 10 известных проблем, восемь из которых были решены более 10 лет назад. Хотя читать все это довольно легко, справедливо будет признать, что хакеры нацеливаются на легкую добычу.

Есть и оправданные случаи, когда обновления безопасности неприменимы обычным способом: например, встроенные системы обычно не предлагают какого бы то ни было управления пакетами. Они также склонны работать на архитектурах не-x86, делая проблематичной компиляцию собственных бинарников. Во врезке (см. *Открытое против закрытого*, внизу) предлагаются инструкции на тот случай, если вы не можете обновить пакеты по стандартным каналам, но это уж действительно последнее прибежище. Просто обновите свою ОС и обновляйте ее постоянно, и жить вам станет намного легче. К тому времени, как вы будете это читать, выйдет Debian Jessie, если вам нужна солидная ОС с долгосрочной поддержкой. Обновив свои старые скрипты/базы данных/вообще всё и избавившись от всех устаревших PHP на своем сайте, вы можете быть совершенно уверены в том, что последующие обновления пакетов, вероятно, не побеспокоят их три последующих года благодаря замораживанию версий программ Debian и применению только отладочной системы безопасности.

Затаившееся вредоносное ПО

Уязвимости способны вызывать цепную реакцию. Например, сомнительный PHP может дать возможность злоумышленнику загрузить свои скрипты на ваш сервер; проблема с Apache может привести к тому, что этот скрипт выполнится и задействует ошибку привилегий где-нибудь еще, которая приведет к тому, что работа будет выполняться от имени root. Тогда ваша машина окажется под контролем злоумышленника, и считайте все свои данные скомпрометированными. Конечно, в теории вы можете этого и не заметить: всё будет отлично выглядеть и работать, но крошечный апплет Flash на вашей домашней страничке отныне

примется потчевать ваших посетителей изумительным коктейлем из вредоносного ПО. Поэтому важно не игнорировать обновления безопасности, даже если уязвимость, которой они адресованы, не предлагает немедленного доступа root. Выгодно будет обзавестись привычкой регулярно и внимательно изучать записи своего системного журнала. Это штука сложная, но есть инструменты, которые вам помогут. Особенно удобный инструмент — *Logwatch*: он умеет проверить доступ к SSH, web, базам данных и любым другим сервисам, с которыми вы работаете, и выдавать результат в удобоваримом формате. Популярный *Awstats* на Perl предлагает привлекательный web-интерфейс для внимательного изучения web, FTP или логов почтового сервера.

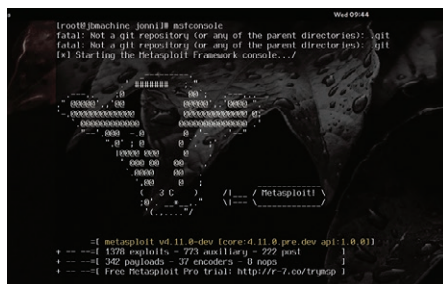
Весьма разумно также присматривать за системной нагрузкой. Команда *uptime* дает вам средние показатели нагрузки CPU за одну секунду, одну минуту и пятнадцать минут, но вы можете вывести график прошлых данных с помощью такого сетевого инструмента, как *Munin*. Программа *vmstat* даст вам информацию по времени ожидания CPU и запросах на подкачку, которые, если их слишком много, указывают на наличие активных операций по вводу/выводу на диске и активному потреблению памяти. Ищите сомнительные процессы. Команда

```
ps awlx --sort=vsz
```

перечислит вам все процессы в порядке возрастания размера, включая разделяемые библиотеки и использование подкачки. Так что всё большее будет перечисляться в конце. Правда, сомнительные программы не обязаны быть большими, а то и (как в случае с руткитом) бывают вообще не видны.

Скрытый руткит

Руткиты — это вредоносные программы, которые используют разнообразные технологии скрытия, чтобы не быть обнаруженными. Они могут прячаться внутри других программ, самого ядра или даже вашего BIOS или встроенных программ других устройств. В последних случаях их вообще невозможно обнаружить, поскольку все системные вызовы, которые в обычном состоянии должны выявить их, могут оказаться нарушенными. Есть такие программы, как *chkrootkit* и *rkhunter*, которые осуществляют проверку на наличие



➤ Metasploit Framework — ценный ресурс для тестов на проникновение, про него даже эта ASCII-корова мычала.

Открытое против закрытого

Широко распространено ошибочное мнение, что поскольку открытый код публичен, он более доступен для атак. Только в 2014 г. был обнаружен возмутительный баг *goto* в библиотеке GnuTLS и уязвимость ShellShock в *Bash* и Heartbleed в OpenSSL. Хотя любой достаточно опытный программист может при подобном факте снисходительно усмехнуться насчет кода, ставшего причиной уязвимостей, это не значит, что ошибки были настолько вопиющими, что их нельзя было не заметить раньше. Читать чужой код довольно трудно, и хотя проекты вроде

OpenSSL все предложения просматривают, всего им не отследить. Возможно, выставлять свое грязное бельё на всеобщее обозрение и бесславно, но, по крайней мере, весь процесс, от обнаружения до устранения, является открытым: вы можете пошутить над непроверенным значением, но можете и одобрительно кивнуть по поводу отлично выполненной отладки.

Любой, кто утверждает, что проприетарное ПО не страдает в такой же степени, должен всего лишь включить компьютер с Windows в первый

вторник любого месяца. В апреле 2015 г. вышло 11 заплаток (четыре из них — критичные), и хотя мы не знаем подробностей, мы видим фразы типа «наращивание привилегий» и «обход системы безопасности», и т.п., и ничего подобного люди на своей ОС видеть не хотят. Такие уязвимости можно также обнаружить с помощью методик типа фаззинга [fuzzing, подача на вход заведомо некорректных значений]. Как только выйдут заплатки, их можно подвергнуть обратному инжинирингу и вооружиться.



По сведениям National Cyber Security Survey, перед CERT-UK стоит задача работы с киберответом на происшествия в Великобритании.

известных руткитов Linux. Вы также можете установить программу обнаружения взлома, например, *AIDE* [см. «По рецептам доктора Брауна», стр. 56 LXF183], которая выявит изменения в вашей файловой системе; но она может потребовать некоторой настройки. Некоторые руткиты и другие зловреды могут полагаться на поддельный модуль ядра. Вы можете принять меры против них, включив в ядре подписи модулей.

Ядро может сгенерировать приватный ключ и сертификат (в котором содержится публичный ключ) за вас, или вы можете использовать собственные. Все последующие модули, которые

«Кто столкнулся с „нулевым днем“, морально обязан сообщить о проблеме.»

вы компилируете, должны быть подписаны этим ключом перед тем, как ядро их загрузит. Удобный скрипт Perl в виде `scripts/sign-file` внутри директории исходников ядра сделает это для вас, при условии, что у вас имеется частный ключ. Например, так подписывается модуль `acx100` (шальной драйвер для определенных беспроводных чипсетов Texas Instruments):

```
$perl /usr/src/linux/scripts/sign-file sha512 /mnt/
sdcard/kernel-signkey.priv /mnt/sdcard/kernel-
signkey.x509 acx100.ko
```

Обратите внимание, как хранятся наши ключ и сертификат на SD-карте. Сертификат публичный, и его можно оставить где угодно, но приватные ключи ни при каких обстоятельствах не должны храниться на тех же носителях, что и данные, которые они защищают. Это всё равно что запереть дверь, но оставить ключ в замке. Как только подписанное ядро будет скомпилировано, вы должны скопировать этот ключ в безопасное место (т.е. вне своей системы) и надежно стереть оригинал. Подпись модулей ядра — это хорошо, но заразиться может и само ядро, и оно позволит загрузить вредоносные модули. Данную проблему можно обойти, загрузив подписанное ядро с EFI, что стоит изучить, хотя это и лежит за пределами темы нашей статьи.

Хэшированные пароли в Linux хранятся в файле `/etc/shadow`, который может читать только root. Если у злоумышленника достаточно ресурсов, он может попробовать взломать эти пароли, чтобы использовать идентификационную информацию

для получения доступа к другим системам. Любые базы данных на скомпрометированной машине предоставляют информацию похитителю — если на машине хранится личная информация, то и она может быть использована, для доступа к другим системам или для атак социальной инженерии. Атакующий может заблокировать вам доступ к вашему компьютеру или просто удалить с него всё, но это его выдает.

Злоумышленник может реализовать на вашем компьютере свои самые большие фантазии. Исследователь безопасности Эндрю Моррис [Andrew Morris] запускает *honeypot* (программу для отлавливания и мониторинга атак), и недавно натолкнулся на злоумышленника, пытавшегося предоставлять и продавать ресурсы одной из заполненных машин в качестве VPS — частного сервера, ха-ха (см. <http://morris.guru/huthos-the-totally-100-legit-vps-provider>). Обычным трюком раньше было установить демон для сбора криптовалюты, но на сегодня результаты получаются ничтожными. А вот уязвимость в программе *DiskStation Manager* (DSM), работающей на устройствах NAS от Synology, привела к тому, что тысячи этих устройств превратились в сборщиков Dogecoin. Полагают, что злоумышленникам удалось собрать более \$600 000. Synology выпустила отладку для DSM в феврале 2014 г., однако массовая атака продолжила сбор средств, так как многие

пользователи поленились внедрить поправку.

Фреймворк Metasploit предоставляет целый массив модулей, позволяющий провести тестирование проникновения через уже известные уязвимости. Например, чтобы найти уязвимости, включенные в CVE в прошлом году, используйте

```
msf > search cve:2014
msf > use auxiliary/scanner/ssl/openssl_heartbleed
... > set RHOSTS targetmachine.com
... > set verbose true
... > exploit
```

Если вас интересует баг Heartbleed (CVE-2014-0160) — пожалуйста:

```
msf > use auxiliary/scanner/ssl/openssl_heartbleed
... > set RHOSTS targetmachine.com
... > set verbose true
... > exploit
```

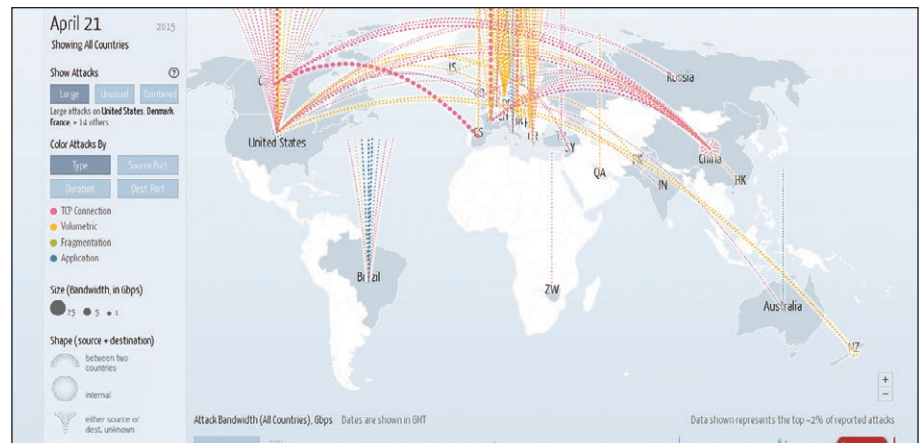
Если существует модуль Metasploit для вредоносного кода, то велика вероятность, что этот вредоносный код кем-то используется, поэтому не пожалейте времени на проверку всех модулей, которые выглядят подходящими к этой ситуации.

Ранее мы вскользь упоминали «эксплоит нулевого дня», не пояснив, что оно такое. Это уязвимость, о которых пока не сообщалось ни публично, ни в частном порядке. В таком случае отладки здесь отсутствуют по определению, и все, что вы можете предпринять — это понадеяться, что вас в ту точку не ударят. В идеальном мире любой, кто столкнулся с нулевым днем, морально обязан сообщить о проблеме соответствующему проекту.

DayZ(ero)

К сожалению, так бывает не всегда — киберпреступники из подпольных сообществ готовы заплатить любую сумму за сходный нулевой день, и маловероятно, что они употребят это знание во благо. Еще больше беспокоит то, что, согласно обнародованным Эдом Сноуденом [Ed Snowden] документам, некоторые правительства (включая правительство США) замараны приобретением и накоплением таких кодов (эксплоитов). Призы за найденные ошибки от Facebook и соревнование *Chrome rwn2own* дают хакерам отличную мотивацию для извещения о найденных уязвимостях, однако проектам с открытым кодом не всегда хватает ресурсов на подобное финансирование. Многие проекты и себя-то с трудом поддерживают: Вернер Кох [Werner Koch] из-за финансовых затруднений чуть было не оставил *GPG*, единственный клиент шифрования с публичным ключом и реально открытым кодом. К счастью, его выручил грант от Linux Foundation и весьма щедрое общественное пожертвование в результате медиа-кампании. Слава Богу, многие разработчики важных проектов Linux имеют работу или поддержку от корпораций.

Это всего лишь беглый обзор безопасности в Linux. Есть масса других проверок, которые можно сделать, и способов действенной защиты... и, к сожалению, много других способов, в результате которых ваш сервер может пасть жертвой атаки. Будьте бдительны, друзья, слушайте советов — и оставайтесь в безопасности. LXF



Кто не верит в реальность DDoS-атак, пусть взглянет на www.digitalattackmap.com и поймет, что неправ.



По советам мистера Брауна

Джолон Браун

В свободное от консультаций по Linux/DevOps время Джолон запускает какой-нибудь стартап. Его главная амбиция — найти причину пользоваться Emacs.

Вопросы пола

Недавно я посетил мероприятие по DevOps (консультанты обязаны вращаться в свете), милое и интересное, но подавляющую часть публики составляли мужчины. На примерно 60 человек пришлось всего три женщины.

Не волнуйтесь, это не очередная колонка, в которой белый мужчина высокопарно рассуждает, «почему в IT нет женщин». Но это не отменяет того факта, что арена DevOps/системного администрирования, похоже, страдает от отсутствия гендерного и прочего разнообразия, несмотря на массу усилий, которые привели к росту разнообразия в IT в целом (включая организации и мероприятия только для женщин и других менее представленных сегментов общества).

Мне кажется нечестным, что женщины (да и другие сегменты общества) упускают более престижные и относительно хорошо оплачиваемые варианты карьеры. Желая помочь, я показал нескольким знакомым женщинам и бывшим коллегам черновик этой колонки и получил несколько отличных идей — да и, что говорить, малость здоровой критики — по поводу того, как лучше это сделать. Основным предложением было менторство, и по различным каналам я предложил потратить немного своего времени на обучение женщин, работающих или желающих работать в сфере DevOps/системного администрирования. Рад сообщить, что кое-кто уже откликнулся!

Говорить об этом открыто тоже важно. Хотя откровенный сексизм больше не почтается в рабочей среде, я работал в тех местах, где женщин уважали (и, как это ни невероятно, оплачивали) меньше, чем мужчин, при той же работе. К сожалению, я никак этому не помешал, но в будущем постараюсь.

Очень важно и образование. Я провел некоторое время на <http://adainitiative.org>, проекте, спонсируемом Google, Red Hat и Linux Foundation, где можно найти массу дополнительной информации по теме.

Тоже решили помочь? Напишите мне на jolyon.brown@gmail.com

Эзотерическое системное администрирование из таинственных закоулков серверной



Контейнеры и компания

Релизы льются рекой: разработчики конфигурационных утилит и PaaS борются за увеличение рыночной доли DevOps.

Свежие данные о рынке от Gartner позволяют предположить, что доля утилит DevOps в 2015 г. достигнет \$2,3 млрд, а к следующему году DevOps «разовьется в одну из главных рыночных стратегий». А потому неудивительно, что за этот более чем обычно насыщенный месяц мы увидели запуск множества продуктов DevOps, контейнеров и облачных продуктов, а также массу пресс-релизов и новостей. PuppetLabs, Chef, Pivotal и даже Microsoft анонсировали новые продукты, запускали новые проекты и улучшали существующие программы. PuppetLabs и Chef (их ежегодная конференция ChefConf прошла в начале апреля) улучшили свои готовые предложения, которые были сосредоточены на автоматизированном управлении, развертывании и тестировании. В обоих анонсированных продуктах обширно представлены контейнеры и облака, и поддержка Docker теперь является обязательной для конкурентов на рынке.

Pivotal официально запустил открытый продукт *Lattice*, призванный помочь в развертывании и управлении контейнерами *Docker*... правильно, в облаке. Приняв за основу свое ПО с архитектурой PaaS [Platform as a Service, платформа как сервис] (Cloud Foundry), *Lattice* позволяет автоматически создавать расписание, балансировать нагрузку, контролировать состояния и управлять журналами. Программа доступна для загрузки на GitHub (и вполне может появиться в будущих выпусках «Советов»). CoreOS (см. след. стр.) анонсировала коммерческую платформу *Kubernetes*, *Tectonic*, которая предлагает поддержку — внимание! — контейнеров Linux, автоматизированного развертывания, документооборота и панелей управления. Microsoft тоже вошла на арену контейнеров, анонсировав контейнеры Hyper-V, которые будут включены в предстоящую Windows 2016 и, что примечательно, будут управляться через *Docker*, что продолжает недавно начавшееся движение компании в сторону некоторого сближения с открытым ПО.

Следующий релиз *OpenStack*, с кодовым названием Kilo, получил релиз-кандидаты, которые вышли перед намеченным на 30 апреля выпуском окончательной версии. Но, к сожалению, не все новости об *OpenStack* позитивны. HP сообщила, что закроет свой публичный облачный сервис на *OpenStack* из-за конкуренции с Amazon и Microsoft. A Nebula, одна из компаний, создававших *OpenStack* и стремившихся к тому, чтобы упростить использование открытой облачной платформы, в апреле закрылась.



CoreOS: Новейшая инфраструктура

CoreOS обещает «динамически масштабируемую и управляемую вычислительную мощность» на манер Google, но так ли это на самом деле?

В открытом ПО и Linux в частности меня очень привлекает их поощрение развития идей. Кто-то где-то находит способ решить некую проблему, делает это, открывает код полученной программы, и с этого момента идея начинает жить сама по себе. Она может расцвести или постепенно угаснуть, может быть поглощена другими идеями (или, напротив, привести к их появлению). Не желая звучать слишком помпезно, скажу, что, на мой взгляд, такое развитие идей и Интернет как проводящая среда для этого являются одними из удивительнейших аспектов современных технологий. Конечно, у такой взрывной креативности и стремления к лучшему есть и недостаток — почти нескончаемый список классных открытых программ для обзора и исследования.

У меня есть огромный список вещей, которые хочется попробовать, и какое-то время в нем находился CoreOS, хотя наверх он никогда не поднимался. Главная причина в том, что у меня пока нет заказчика, которому очень бы хотелось ее попробовать (проект появился только в 2013 г.), и CoreOS приходилось торчать в очереди. Но тут *Linux Format* опометчиво решил заплатить мне за данную статью [Ред.: — Э, а мы что, правда ему платим?], и у меня появился удобный повод.

Время CoreOS пришло, и теперь я удивляюсь, чего же я ждал так долго. На сайте CoreOS описана система, которая ставит много галочек в моих интересах как инженера DevOps/системного администратора, обещая мне систему, нацеленную на автоматизацию, простоту развертывания приложений, безопасность, надежность и стабильность.

При тестировании таких вещей я обычно выдвигаю сценарий, способный возникнуть у потенциального заказчика, чтобы лучше понять, как справляться с подобными случаями. Поэтому в этой и следующей статье я намерен оценить жизнеспособность CoreOS как варианта для воображаемого заказчика, который хочет перевести на нее свой хостинг со сторонними сайтами. Допустим, заказчик ищет способ снижения расходов на инфраструктуру и думает, что набор функций CoreOS идеально подошел бы для данной цели, и поэтому рассматривает те возможности, которые станут хорошей отправной точкой.

Проникаем в суть

CoreOS основана на ChromeOS/*Chromium* — да, браузерной ОС от Google — которая, естественно, базируется на ядре Linux

(фактически сборка основана на Gentoo). Но задача ее не в том, чтобы предоставить набор возможностей обычного настольного дистрибутива — CoreOS сосредоточена исключительно на управлении и развертывании приложений и минимизацией всех прочих работ, обычно связанных с такими задачами, как внесение правок, обслуживание, перемещение нагрузки между системами при выходе одной из них из строя и т.д.

Прежде всего, иссохших системных администраторов типа меня в CoreOS поражает то, что здесь нет традиционных систем управления правами, таких как *yum* или *apt* (рядовой случай, учитывая, что я совсем недавно научился понимать все параметры этих команд). Вместо этого CoreOS предлагает автоматические обновления, загружаемые с сайта проекта, по аналогии с тем, как работает браузер *Chrome*.

Во-вторых, в сердце CoreOS лежит облегченный хост управления. Приложения запускаются на этом хосте как на гипервизоре, но вместо виртуальных машин в CoreOS используется *Docker*, который запускает приложения в контейнерах. С помощью утилиты под названием *Fleet* CoreOS распределяет контейнеры по кластеру компьютеров и может применять правила соответствия (т.е. «Не запускать все компоненты моего высокодоступного приложения на одном и том же физическом хосте»).

Наконец, в CoreOS есть *etcd* — программа обнаружения сервисов. Она позволяет хранить настройки вне приложений — больше не придется жестко кодировать строки подключения к базам данных и т.п. — а также поддерживает репликацию настроек на любой созданный нами кластер. Интригует, да? Приступим же к тестированию.

Настройка тестовой системы

CoreOS поддерживает разнообразные цели для развертывания. Как и следовало предполагать, была предпринята масса усилий для того, чтобы популярные облачные провайдеры (Amazon, Rackspace, Azure, Google Compute Engine, Digital Ocean) заработали сразу же; также можно воспользоваться *OpenStack* и *Vagrant*. Раз я пытаюсь оценить это с точки зрения клиента, то локально устанавливаю *Vagrant* на ПК с Linux.

В документации CoreOS рекомендуется использовать версию *Vagrant* выше 1.6.3, поэтому я воспользуюсь Deb-файлом прямо с www.vagrantup.com для установки в чистую Ubuntu 14.04, где по умолчанию используется более ранняя версия. Я также

»

CoreOS + Kubernetes = Tectonic (но не открытый)

Перед самой отправкой этого номера в печать CoreOS анонсировала новые инвестиции от Google Ventures и запуск нового сервиса: *Tectonic*. Это коммерческий продукт (CoreOS предлагает управляемую поддержку по аналогии с Red Hat и т.п.), который включает *Kubernetes*, открытый проект от Google.

Kubernetes — это система управления для контейнеров *Docker*, которая представляет их в виде узлов вычислительного кластера и активно управляет ими.

Интересный момент для фанатов инфраструктур (вроде меня) в том, что этот продукт основан на коде внутренней инфраструктуры Google (на котором работают такие сервисы как поиск, Gmail и т.д.). Учитывая репутацию поискового гиганта (а также влияние на данную сферу его многолетних научно-исследовательских разработок) и таинственную ауру, окружающую огромные дата-центры Google, на их внутреннюю структуру стоит взглянуть.

CoreOS приняла решение не открывать код своей реализации, что печально, хотя компания

утверждает, что все существующие компоненты CoreOS останутся неизменными. Подчеркнем, что *Tectonic* будет собственным брендом компании, отдельным от CoreOS.

Но *Kubernetes* — не единственный игрок на рынке. На него похожа *Apache Mesos*, а *Docker* привел к появлению достаточно большого количества проектов PaaS, таких как *Deis*. Поскольку *Kubernetes* сам по себе не открыт, уверен, мы увидим и другие попытки, а открытое ПО имеет привычку в итоге побеждать.

установил *VirtualBox* для запуска виртуальных машин. После этого нужно клонировать git-репозиторий CoreOS с *vagrant*:

```
git clone https://github.com/coreos/coreos-vagrant.git
cd coreos-vagrant
```

Заглянув в результирующий каталог **coreos-vagrant**, вы найдете там несколько созданных файлов-примеров **.sample**. Первым делом надо взглянуть на **user-data.sample** (и я немедленно переименую его в **user-data**). Этот файл содержит следующую строку (изначально закомментированную):

```
#discovery: https://discovery.etcd.io/<token>
```

Для начала мне нужен пример токена от публичного сервиса CoreOS (экземпляр *etcd*, доступный через Интернет), расположенного на <https://discovery.etcd.io/new>. Зайдя на него в браузере (или из *curl*), вы получите строку с токеном в конце, которую я вставлю в файл **user-data**, тем самым ее раскомментировав:

```
discovery: https://discovery.etcd.io/d72db3274bbfcff3069683ba649e6c69
```

Этим публичным сервисом обнаружения можно воспользоваться вместо настройки собственного, но стоит отметить, что многие клиенты не смогут (вероятно, из-за политик) или не захотят положиться в этом вопросе на третью сторону. Настройке собственного сервиса я посвящу одну из будущих статей.

Потом я переименую **config.b.sample** в **config.rb**, открою его и изменю следующие строки:

```
#$num_instances=1
...
#$update_channel='alpha'
$num_instances=3
...
$update_channel='stable'
```

Я изменил количество запускаемых экземпляров на 3, так как позднее хочу протестировать кластеризацию с *Fleet*. Я нахожусь на самом начальном этапе тестирования, поэтому выберу стабильный релиз CoreOS, чтобы невзначай не потратить несколько часов на погоню за потенциальными ошибками альфа-релиза, о которых ничего не знаю. Если я обнаружу, что необходимые мне возможности есть только в релизах для разработчиков, я приму этот вызов.

Пора запустить систему. Отойдите назад, взмахните волшебной палочкой... или хотя бы наберите

```
vagrant up
```

Поскольку я делаю это в первый раз, *Vagrant* загрузит образ виртуальной машины для CoreOS и всё запустит. Через несколько минут я смогу запустить

```
vagrant status
```

и получить следующий результат:

```
Current machine states:
core-01 running (virtualbox)
core-02 running (virtualbox)
core-03 running (virtualbox)
```

Затем перейду к команде

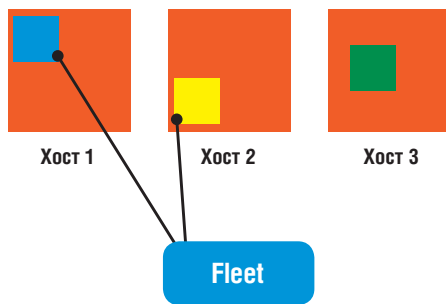
```
vagrant ssh core-01
```

которая позволит мне войти в первую виртуальную машину. Беглый взгляд вокруг подтверждает, что это Linux, хотя и слабозагруженный. Взгляд на список процессов с помощью **ps** показывает, что запущены *Systemd*, демон *Docker* и нечто под названием *update_engine*, и на этом почти все.

Изучение правок к CoreOS

Пока все идет нормально. Первая задача в моем списке — исследовать эффективность механизма внесения правок в CoreOS, так как этот вопрос будет важным для моих заказчиков. В экземплярах *Vagrant* работает CoreOS версии 607.0.0 (это последний

Приложения



➤ *Fleet* поддерживает распределение контейнеров по кластеру CoreOS и умеет заботиться о правилах соответствия, перемещать контейнеры на время обслуживания и запускать правильное количество контейнеров.

etcd — сервис настройки и обнаружения

Согласно сайту CoreOS, «*etcd* — открытое распределенное хранилище пар ключ–значение, которое предоставляет распределенную настройку и обнаружение сервисов для кластеров CoreOS». Но что именно это означает? Приложения (в контейнерах), запускаемые в CoreOS, могут считывать и записывать данные в *etcd*. Популярный пример таких данных — информация о подключении к базе данных. Для реализации можно воспользоваться простым API HTTP или удобной командой (**etcdctl**). В тестовой системе из нашего урока можно запустить

```
etcdctl set /xf hello
```

на одной виртуальной машине, а затем

```
etcdctl get /xf
```

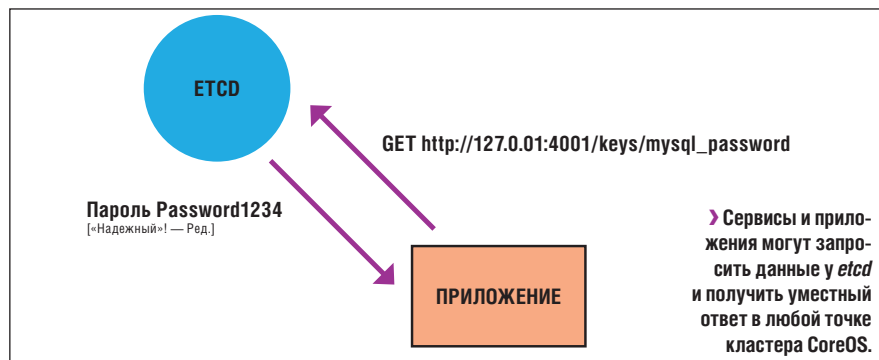
на другой, чтобы получить ту же строку. Это означает, что мы можем обращаться к данным из любого узла кластера, и данные всегда будут синхронизированы.

Для подобного обнаружения сервисов нужно заставить их регистрироваться в *etcd* при запуске. Другие системы могут проверять обновления и изменять себя сами: например, балансировщик

нагрузки регистрирует новые серверные сервисы без взаимодействия с системным администратором. *etcd* высокодоступна: она выдерживает выход из строя нескольких узлов (но всегда требует кворума систем для работы во избежание дробления вычислительных ресурсов). Существует ряд архитектурных решений, применимых при разрастании

кластера *etcd*: например, воспользоваться прокси для обработки возросшего трафика.

etcd — не единственный проект, решающий подобные задачи. Есть, например, *Zookeeper* (изначально от Yahoo!, но ныне проект *Apache*), *Consul* и *Doozerd*. У всех свои плюсы и минусы, и сфера разработки инфраструктур бурно разрастается.



➤ Сервисы и приложения могут запросить данные у *etcd* и получить уместный ответ в любой точке кластера CoreOS.

стабильный релиз, согласно сайту <http://coreos.com>). Для работы с обновлениями в CoreOS есть два корневых раздела: **USR-A** и **USR-B**. Сначала — при свежей установке — система загружается в первый из этих разделов, который монтируется только для чтения в **/usr**. Утилита обновления устанавливает новую копию операционной системы в альтернативную файловую систему **/usr** (которая изначально пуста, ради экономии места). После перезагрузки система загружается с использованием свежееустановленной версии ОС. Если в ней что-то пойдет не так, систему можно перезагрузить, и произойдет откат к предыдущей версии.

Этот подход к внесению правок очень похож на некогда популярный в старых системах Solaris с комплектом дисков, а также в системах Linux с зеркалируемой загрузкой, до того, как распределенной стратегией внесения правок стали моментальные снимки [snapshots]. Но большое отличие, разумеется, в том, что все это делается автоматически и не является эквивалентом развертывания. Уверен, что у более крупных заказчиков, имеющих сотрудников по обслуживанию обновлений, возникнут дискуссии о готовности доверить так много третьей стороне, но на мой взгляд это дает реальную экономию времени. Согласно сайту CoreOS, утилита обновления регулирует свою полосу пропускания и процессорное время, что позволяет избежать прерывания или ухудшения качества работы сервиса.

Решив посмотреть на разделы своей системы, я воспользовался командой **cgpt**, которая послушно «выплюнула» порцию информации (здесь мы ее сократили для экономии места).

```
sudo cgpt show /dev/sda
270336 2097152 3 Label: "USR-A"
Type: Alias for coreos-roots
UUID: 7130C94A-213A-4E5A-8E26-6CCE9662F132 Attr: priority=1
tries=0 successful=1
2367488 2097152 4 Label: "USR-B"
Type: Alias for coreos-roots
UUID: E03DD35C-7C2D-4A47-B3FE-27F15780A57C Attr: priority=0
tries=0 successful=0
```

Заглянув в каталог **/usr/boot/grub**, я могу найти соответствующие записи для USR-A и B в файлах **menu.lst**. Это решение кажется мне довольно элегантным, так как я вижу, что собирается сделать утилита обновления. Поскольку эта ОС основана на *Systemd*, я смогу воспользоваться командой **journalctl** [подробнее об этом см. «Учебники», стр. 64]. Последующая команда вывела несколько милых строк лог-файла, которые сообщили мне, что ОС не смогла разрешить хост public-update.core-os.net из-за проблемы с подключением. Строки также поведали, что процесс обновления основан на проекте Omaha Google.

```
journalctl -f -u update-engine
```

Должен сказать, на меня это решение произвело довольно сильное впечатление. Так как CoreOS настаивает, чтобы все приложения запускались в контейнерах, то применение заплаток к соответствующей ОС с помощью этого подхода существенно разделяется на части. Хотелось бы видеть, насколько это управляемо в рабочей системе и можно ли настроить локальные репозитории для ультрабезопасных сред, но для нашего гипотетического заказчика это, наверное, именно то, что надо.

Запускаем Fleet

Теперь я хочу рассмотреть *Fleet*, менеджер кластера. *Fleet* работает с модулями *Systemd* (другой компонент *Systemd*, также известный как цели). Модули — это конфигурационные файлы, описывающие свойства процессов, и в моем случае они будут сильно привязаны к *Docker*. В CoreOS файлы модулей находятся в каталоге **/etc/systemd/system**.

Для просмотра состояния кластера *Fleet* можно воспользоваться командой **fleetctl**:

```
fleetctl list-machines
```



► При запуске или остановке контейнера он обновляет свой статус в *etcd*, и другие системы могут выполнить действия в соответствии с этим (например, сервисы регистрации балансировщика нагрузки, которые могут включаться и отключаться).

Эта команда покажет три виртуальных машины Vagrant в нашем кластере:

```
MACHINE IP METADATA
c9fc38db... 172.17.8.101 -
ca1115a3... 172.17.8.103 -
e7713580... 172.17.8.102 -
```

Во время тестирования не обошлось без проблем. Все они сводились к ошибкам в файле **user-data**. Один раз я по ошибке переименовал его в **user.data** (пропустив дефис), и *Vagrant* не смог скопировать его в должный каталог (**/var/lib/coreos-vagrant/vagrantfile-user-data**). В другой раз я случайно затер ключевое слово “discovery:”, что привело к ошибкам *Yaml*. Вот несколько полезных команд, которые помогут выявить такие ошибки:

```
systemctl status -l fleet
journalctl -b -u fleet
```

В моей системе все хорошо, поэтому теперь я хочу запустить тестовый сервис. На сайте *Docker* есть пример очень простого тестового контейнера, который я смогу адаптировать здесь. На своей первой виртуальной машине *Vagrant* я создал следующий файл и назвал его **testapp.service**:

```
[Unit]
Description=TestApp
After=docker.service
Requires=docker.service
[Service]
TimeoutStartSec=0
ExecStartPre=-/usr/bin/docker kill testubuntu
ExecStartPre=-/usr/bin/docker rm testubuntu
ExecStartPre=/usr/bin/docker pull ubuntu
ExecStart=/usr/bin/docker run --name testubuntu ubuntu /bin/sh -c "while true; do echo Hello Linux Format; sleep 1; done"
ExecStop=/usr/bin/docker stop testubuntu
```

Я могу запускать и отслеживать этот файл с помощью команд **fleetctl**:

```
fleetctl start testapp.service
fleetctl list-units
И в конце концов меня вознаградил работающий сервис!
core@core-01 ~ $ f
UNIT MACHINE ACTIVE SUB
testapp.service c9fc38db.../172.17.8.101 active running
```

Довольно просто, не так ли? В следующей статье я рассмотрю *Fleet* и *Docker* в CoreOS гораздо более подробно и попробую выполнить репликацию, которая может понадобиться моему заказчику: эластичный сервис хостинга. Увидимся! **LXF**

ПОДПИШИСЬ НА LINUX FORMAT!

Оформи подписку на печатную версию журнала Linux Format на 2015 год и получи в подарок диск с архивом номеров 2005—2014, а также подписку на электронную версию в формате PDF



SHOP.LINUXFORMAT.RU

Варианты доставки

- » Почтой по России простой или заказной бандеролью
- » Курьерской службой СПСР или IML
- » В виде PDF-файлов для подписчиков электронной версии

Редакция

- » Санкт-Петербург: пр. Медиков, 5, корп. 7, тел. (812) 309-0686
- » Москва: тел. (499) 271-4954

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Неспешная эволюция — это благо

Труд сделал из обезьяны человека.
Фридрих Энгельс

Тихо и незаметно вышло очередное ежегодное обновление флагманского дистрибутива *LaTeX* (компьютерная система верстки текстов), а именно *TeX Live 2015*.

На самом деле, не появилось ничего особенного. Юникодные названия и определения внедрили непосредственно движок *LaTeX2ε* — ранее всё это было доступно через расширения. Добавили расширенную поддержку JPEG в *pdfTeX* (в частности, *Exif* и *JFIF*). Были улучшены экспериментальные движки *LuaTeX* и *XeTeX*. *MetaPost* заговорил по-японски. *MacTeX* поднял свой китайский. *fmtutils* в очередной раз переписан заново.

Если не хочется ждать, пока изменения проявятся в стандартном репозитории вашей платформы, то дистрибутив можно скачать прямо с сайта www.tug.org/texlive/ и установить его локально на два десятка поддерживаемых из коробки платформ или собрать его из исходников.

Да, время от времени всплывают ошибки (куда же без них), но в большинстве своём всё работает как ожидается и как должно. Никаких революций, и это позволяет жить спокойно, сосредоточившись на написании общественно полезных текстов.

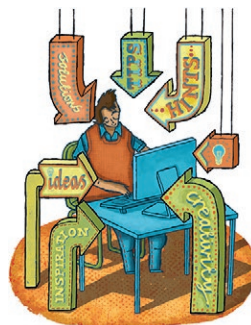
PS Обновился *GNU MediaGoblin*. Этот софт позволяет устроить свой локальный “YouTube”, хотя и придётся потрудиться.
E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Наводить статистику 60

Шон Конвэй не любит, когда система тормозит, и всем задает вопрос, как с этим бороться. Да сам же на него и отвечает (с помощью *sysstat*).



Читать журналы 64

Оказывается, в журнале *Systemd* масса интересной информации! Но поскольку ее не удастся прочесть без *journalctl*, следует обратиться к содействию **Нейла Ботвика**.



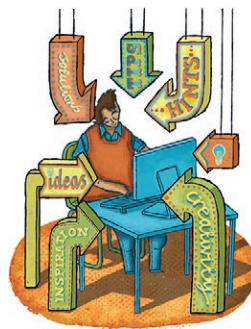
Выдавать права 66

Восседая на троне **root**, **Нейл Ботвик** милостиво раздает пользователям привилегии на управление частями системы. А иногда учиняет им групповуху.



Следить за системой 68

Инструментарий для работающего ядра? Заявка серьезная. Но у **Михалиса Цукалоса** все получится — ведь он разобрался с *SystemTap*.



Ломать роутеры 72

Вслед за **Мэттом Билби** погрузимся в мир нестандартных прошивок и конфигураций. Но не забывайте: вы действуете на свой страх и риск, и прощай, гарантия!

Код в учебниках

Строки исходного кода помещаются в цветные плашки. Если строка кода не умещается 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 неведомое

Inotify и getops 76

Д-р **Крис Браун** анализирует оповещения системы и аргументы командной строки.

Scala 80

Михалис Цукалос объясняет базовые концепции языка функционального программирования.

Unity 5 84

Обзор Unity 5, **Робин де Йонг** не остановился на достигнутом и написал еще и руководство.

Sysstat: Как работает?

Шон Конвэй ставит перед сисадмином вопрос о производительности и сам же на него отвечает, благодаря утилите *sysstat*.



Наш эксперт

Шон Конвэй — технический специалист; он отцеживает из своего 30-летнего опыта живительные стаканы практических знаний, помогающих системным администраторам отполировать свои навыки.

Сначала поставим вопрос на засыпку, который частенько задают кандидатам на должность системного администратора во время собеседования: «Вы получили от пользователей отчет с сообщением, что поддерживаемая вами база данных работает очень медленно. Какие действия следует предпринять?»

Наш урок поможет вам получить на это ответ (и, возможно, работу). Приоткрыта будет лишь часть всего ответа (чтобы получить работу, вам придется попотеть дополнительно). Мы прольем свет на то, как добыть информацию об использовании процессора, памяти и ввода/вывода с помощью встроенной метрики ядра Linux под названием *sysstat*, взяв за основу дистрибутив CentOS.

Sysstat — набор утилит, позволяющих просмотреть счетчики ядра Linux. По счетчикам системный администратор может составить представление о производительности. Цель — сузить с помощью счетчиков ядра диапазон поиска проблемы.

Подготовимся

В ядре Linux есть встроенные счетчики, предоставляющие данные об активности системы. Они записывают статистику ввода/вывода, такую как скорость передачи данных на диски, страницы памяти, информацию о процессах, прерываниях, сети, использовании подкачки, статистику ядра и терминалов. Данные о процессорах записываются глобально или по отдельным процессорам, и пакет *sysstat* предоставляет средства для создания отчетов по данным счетчиков.

Наш дистрибутив — CentOS, но утилита не ограничена Red Hat-версиями Linux и доступна также в дистрибутивах, производных от Debian. Чтобы установить ее в Debian, скомаундите

```
sudo apt-get install sysstat
```

а на платформе Red Hat —

```
sudo yum install sysstat
```

или

```
sudo rpm -Uvh sysstat-11.0.0-3.fc21.x86_64.rpm
```

Список утилит, входящих в пакет *sysstat*, покажет команда **rpm -ql sysstat**:

» **-/usr/bin/sar** Собирает, выводит и сохраняет данные об активности системы.

» **-/usr/lib64/sa/sadc** Собирает данные об активности системы.

» **-/usr/lib64/sa/sa1** Собирает и сохраняет двоичные данные в файлах ежедневной активности системы в **/var/log/sa**.

» **-/usr/lib64/sa/sa2** Создает отчет в удобном виде из файла ежедневной активности системы в **/var/log/sa**.

» **-/usr/bin/sadf** Отображает данные, собранные *sar*, в других форматах (например, CSV, XML).

» **-/usr/bin/iostat** Выводит статистику по использованию процессора и ввода/вывода для устройств и разделов.

» **-/usr/bin/mpstat** Выводит статистику по использованию процессора для каждого доступного процессора.

» **-/usr/bin/pidstat** Выводит статистику по задачам, привязанным к идентификатору процесса.

» **-/usr/bin/nfsiostat-sysstat** Выводит статистику ввода/вывода для NFS.

» **-/usr/bin/cifsioat** Выводит отчет, состоящий из заголовка CIFS, за которым следует статистика для каждой смонтированной файловой системы CIFS.

При установке очередного пакета выполняются изменения в файле *crontab* для настройки цикла сбора данных, который каждый день начинается с нуля, а информация выводится в отчетах ASCII.

```
cat /etc/cron.d/sysstat
# Run system activity accounting tool every 10 minutes
*/10 * * * * root /usr/lib64/sa/sa1 1 1
# 0 * * * * root /usr/lib64/sa/sa1 600 6 &
# Generate a daily summary of process accounting at 23:53
53 23 * * * root /usr/lib64/sa/sa2 -A
```

Мы описали установку кратко, чтобы хватило места на рассказ о трех командах утилиты: **sar**, **sa1** и **sa2**. Кому нужны подробности установки, советуем зайти в блог <http://sebastien.godard.pagersperso-orange.fr>.

Работаем с sar

Главная команда пакета *sysstat* — **sar**. Она сообщает об активности системы. Команда извлекает содержимое внутренних счетчиков ядра Linux и записывает данные в стандартный вывод (например, на экран).

Команда **sar** поддерживает флажки-опции параметров и параметры. Опции **sar {-flag} <interval> <count>** используются для указания набора счетчиков. Параметры **<interval>** и **<count>**, следующие за опциями, используются для указания того, когда и как часто следует выводить данные.

Следующий набор команд создаст 60-секундный отчет путем изменения параметров **interval** и **count**. Опция **-A** означает, что запрашиваются все счетчики:

```
/usr/bin/sar -A 5 12 # =60sec (5secX12count)
/usr/bin/sar -A 10 6 # =60sec
/usr/bin/sar -A 20 3 # =60sec
```

Используя один флажок или сочетая 18 доступных флажков, можно получить значения конкретных счетчиков. Помните, что счетчики ядра Linux содержат информацию об использовании процессора, памяти и ввода/вывода. На DVD этого номера можно найти таблицу опций, сгруппированную по категориям, и список имен файлов, возвращаемых при выполнении каждой команды. На map-страницах команды **sar** представлено описание каждого

Скорая помощь



Чтобы найти файлы библиотек *sysstat* в 32-битном Linux, удалите "64" из пути каталога.

Если б не люди...

Все сисадмины признаются, что не будь пользователей, система работала бы отлично — но тогда они были бы безработными. Пользователи предоставляют информацию о проблеме. Сетевая часть включает все компоненты, поддерживающие соединение, от хоста до сервера, обеспечивающего приложение. Программную часть решения проблемы

можно разделить на операционную систему, базу данных и приложение. К аппаратной части относятся процессор, оперативная память и ввод/вывод. Сбор полезной информации из отчетов пользователей для решения проблем производительности бывает сложной задачей. Первым делом следует выяснить, с чего проблема началась. Ядро имеет

несколько счетчиков, которые хранят информацию об использовании процессора, памяти и ввода/вывода с течением времени. Зная, когда возникла проблема, и обладая информацией счетчиков ядра, системный администратор потенциально может связать проблему производительности с аппаратной ошибкой.

из полей таблицы. Некоторым читателям удобнее будет читать онлайн-версию: <http://linux.die.net/man/1/sar>.

Команда **sa1** собирает данные счетчиков ядра и сохраняет их в ежедневном двоичном файле `/var/log/sa/sa#`. Другая команда пакета, **sa2**, преобразует двоичный файл в человеко-читаемый, в формате ASCII, который хранится в `/var/log/sa/sar#`. **sa2** понимает опции и параметры команды **sar**. Отчеты в формате ASCII можно генерировать для конкретных счетчиков с помощью тех же флажков, что используются с командой **sar**. Примечание: символ решетки #, добавляемый к именам файлов, заменяется на день месяца. Конфигурация `crontab`, которую мы описали при установке `sysstat`, настраивает **sa1** на ежедневный запуск. Затем `cron` запускает **sa2** для преобразования двоичных данных, собранных **sa1**, в отчет в формате ASCII. Эти задания формируют список дневных счетчиков, с помощью которых системный администратор может проанализировать проблемы производительности при их возникновении. Обратите внимание, что параметры `<interval>` и `<count>`, используемые в `crontab`, имеют не слишком большой шаг изменения. Один из параметров `crontab` закомментирован. Удалив комментарий, при необходимости можно получить более надежные счетчики.

Запуск утилиты `sysstat` как задания `cron` помогает регулярно собирать и сохранять значения счетчиков. При возникновении

проблемы сохраненные счетчики можно проанализировать для ее решения; к тому же **sa1** — отличный способ сбора информации о счетчиках при тестировании программ. Скажем, администратор проверяет новую процедуру запроса к базе данных. Запуск **sa1** во время тестирования базы данных позволит получить отчет, который поможет понять, было ли какое-либо внешнее воздействие при выполнении кода базы данных.

Следующая команда (и далее еще три) используются для установки десятиминутного интервала сбора данных отчета. Первая вставляет пустую запись в файл счетчика, чтобы гарантировать сброс счетчиков в ноль:

```
usr/lib64/sa/sa1 --boot
```

Помните, что если **sa1** запускалась через `crontab`, она должна сохранить счетчики. Эта команда обеспечивает исходное состояние для тестирования. Во время тестирования стоит отключать задание `cron` на сбор счетчиков — для обеспечения целостности данных. Вторая команда обеспечивает сборку каждые 10 секунд и 60 раз, что эквивалентно 10 минутам сбора данных:

```
usr/lib64/sa/sa1 10 60
```

Третья команда создает файл в формате ASCII, а четвертая выводит его в стандартный поток вывода. К имени файла добавится суффикс в виде дня месяца.

```
usr/lib64/sa/sa2 -A
```

Скорая помощь

Сбор данных со счетчиков — дополнительная нагрузка на систему. Лучше сохранять набор счетчиков минимальным, а увеличивать только при необходимости.

Данные ядра о сети

Время	Интерфейс	Прием (пакетов в секунду)	Передача (пакетов в секунду)	Прием (кБ/с)	Передача (кБ/с)	Прием (mcast/c)
12:46:09	eth0	9,02	2,81	2,12	0,78	0,6
12:46:14	eth0	3509,18	1758,08	5190,03	122,25	0,2

Целиком таблица приведена на <http://pastebin.com/RzUtbXzF>.

Время	Интерфейс	Прием (ошибок в секунду)	Передача (ошибок в секунду)	Конфликтов в секунду	Прием (потерь в секунду)	Прием (кадров в секунду)
12:46:09	eth0	0	0	0	0	0
12:46:14	eth0	0	0	0	0	0

Время	Интерфейс	Прием (пакетов в секунду)	Передача (пакетов в секунду)	Прием (кБ/с)	Передача (кБ/с)	Прием (mcast/c)
13:38:54	eth0	6314,95	3003,23	9351,52	210,38	0,4
13:38:59	eth0	6400,6	2972,95	9480,99	208,45	0,4

Время	Интерфейс	Прием (ошибок в секунду)	Передача (ошибок в секунду)	Конфликтов в секунду	Прием (потерь в секунду)	Прием (кадров в секунду)
13:38:59	eth0	0	0	1545,29	12,22	0
13:39:04	eth0	0	0	1512,05	7,63	0

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!

```
less /var/log/sa/sar#
```

Примечание: команда **sa2** использует флажок **-A**. Его можно заменить любым из флажков параметров (см. таблицу на **LXFVDV**), чтобы получить файл ASCII, содержащий значения заданных счетчиков.

Игравшие в футбол по левую сторону Атлантики знают: если тренер велел что-то сделать, отнюдь не факт, что это произойдет. Если вы полагаете, что собрав значения счетчиков, останется только посмотреть на их значения и найти те, которые превышают лимиты, спешу разочаровать: все не так просто. Без некоторого опыта в придании знаниям нужной формы вам не достичь цели.

Просмотр значений отдельных счетчиков — хороший способ локализации проблемы. Понимание взаимодействия счетчиков позволяет оценить работу всей системы, а не только одной

ее части. В следующем разделе мы воспользуемся утилитами из **sysstat** для сбора данных, запустив скрипт, который должен повлиять на производительность системы. Собранные данные можно проанализировать и лучше понять, что происходит в системе, основываясь на значениях системных счетчиков.

Поскриптим

Один скрипт повышает нагрузку на центральный процессор. Он максимально задействует все четыре ядра нашего процессора в тестовой системе. Другой скрипт занимает участки свободной памяти до тех пор, пока ее не останется — прямо как акула, плывущая вдоль пляжа, подъезжает пловцов в фильме «Челюсти». Последняя порция данных — из реального примера — счетчик ядра для сетевой карты неправильно настроенной системы.

Использование процессора и памяти

Время	Процессор	%use	%system	%iowait	%ПО	%idle
12:51:20	весь	0,05	0,2	0	0	99,75
12:51:20	0	0	0	0	0	100
12:51:20	1	0	0	0	0	100
12:51:20	2	0,2	0,8	0	0	99
12:51:20	3	0	0	0	0	100
12:51:25	весь	0,05	28,76	0	0	71,19
12:51:25	0	0	38,6	0	0	61,4
12:51:25	1	0	18,64	0	0	81,36
12:51:25	2	0,2	19,14	0	0	80,65
12:51:25	3	0	38,6	0	0	61,4

Перезануск linux (<http://pastebin.com/f4PNYSVA>)

Время	Процессор	%user	%system	%iowait	%ПО	%idle
09:35:22	весь	0,57	9,34	19,81	0,6	69,68
09:35:22	0	0,4	9,76	35,96	0,94	52,94
09:35:22	1	0,27	7,56	9,02	0,93	82,23
09:35:22	2	1,73	19,65	34,26	0,53	43,82
09:35:22	3	0	0,53	0	0	99,47
09:35:25	весь	0,3	5,69	49,11	0,44	44,46
09:35:25	0	0,59	7,12	81,31	0,89	10,09
09:35:25	1	0,3	6,23	25,22	0,3	67,95
09:35:25	2	0,29	9,41	56,76	0,59	32,94
09:35:25	3	0	0	33,33	0	66,67

Статистика использования памяти (<http://pastebin.com/cfa8UPuB>)

Время	kb mem free весь	kb mem used	% занятой памяти	Кэшировано кб	кб commit	% commit
09:35:14	1689104	230812	12,02	13600	150032	2,48
09:35:22	59576	1860340	96,9	10968	2323400	38,41
...						
09:36:20	53104	1866812	97,23	9492	5694032	94,14
09:36:26	1761968	157948	8,23	4224	101548	1,68

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe/

Данные счетчиков собирались и отображались с помощью описанных ранее четырех команд (см. стр. 61 внизу и стр. 62 внизу), и, само собой разумеется, системный администратор должен попрактиковаться в применении утилит пакета *sysstat*. Так он познакомится с выводом команды и узнает, как употреблять ее наиболее эффективно. Использование команд для тестирования — простой способ сочетания практики с получением результатов.

Объем данных, которые можно привести в этой статье, ограничен, но чтобы у вас была вся информация для дальнейшего изучения утилит, мы привели полный набор данных для этих сценариев на **LXF DVD** или на URL, указанных в таблицах.

Тестовые сценарии

В первом тестовом сценарии используются процессорные ресурсы (см. файл **CPU_loaded.txt** на **LXF DVD** или стр. слева). В полях `%user` и `%system` отображается количество процессорного времени в режиме пользователя и в режиме системы. Столбец `%idle` сообщает о частоте простоя процессора. Поле `%iowait` — это время, которое процессор тратит на выполнение запросов ввода/вывода. Время в столбце `%idle`, близкое к нулевому, означает, что процессор является узким местом системы, а высокие значения `%iowait` свидетельствуют о возможных проблемах с диском. Просмотрев дополнительные данные на диске, вы увидите, как влияет выполнение скрипта на загрузку ядер.

Отчет об этих счетчиках после запуска скрипта обращения к памяти также означает, что ресурсы процессора используются. Значения в поле `%idle` низкие. Также обратите внимание на поле `%iowait`. В последнем сценарии это поле не содержит данных, т. к. скрипт направлен только на процессор. Этот сценарий предполагает, что нам стоит взглянуть поглубже, не торопясь относить эту проблему к загрузке процессора.

Взглянув на отчет о статистике использования памяти (см. файл **memory_memeater.txt** на **LXF DVD** или стр. слева), вы захотите сосредоточиться на полях `kbmemfree` и `kbmemused`, которые соответствуют свободной и используемой памяти. Обратите внимание на серьезное уменьшение счетчиков для `kbmemfree` (свободная память). Кто-то отхватил большой кусок памяти и затем его освободил. Теперь у нас есть дополнительная информация для объяснения изменения показателей процессора (`%idle`).

На **LXF DVD** в файле с именем **memory_memeater.txt** вы найдете статистику использования файла подкачки. В таблице с данными взгляните на поле `%swrused`. Нулевое значение в этом поле означает, что файл подкачки не используется.

Система начинает использовать подкачку, когда ей нужно больше памяти. При этом она записывает на жесткий диск наименее часто используемые страницы физической памяти. Обращение к файлу подкачки (т. е. диску) выполняется гораздо медленнее, чем к оперативной памяти.

На пересылку данных из подкачки в оперативную память уходит время. Учащение использования подкачки негативно влияет на производительность. Этот маленький скрипт, использующий память, определенно пощекотал счетчики ядра.

Давайте вновь рассмотрим собранные данные в свете того, что мы узнали. Скрипт в процессе работы занимает 1024 Кб блоков свободной памяти. Поскольку свободная память занята, начинает использоваться файл подкачки. Поскольку нужно больше свободной памяти, используется больше файла подкачки. Процессор выполняет какие-то вычисления, память используется, и увеличивается использование файла подкачки, что оказывает негативное влияние на производительность.

Не отчаивайтесь, если ваша голова уже стала квадратной. Решение проблем с производительностью системы — та еще история. Чтобы правильно анализировать данные и делать какие-то выводы, нужны практика и опыт.

Последний сценарий — это не скрипт, а настоящая ошибка. Данные счетчиков собирались в неправильно настроенной системе. Надеюсь, анализ данных этого сценария поможет вам восстановить веру в эффективность утилит *sysstat*, если она вдруг подверглась сомнению.

Временная отметка 12:46:14PM (см. файл **nic_sar_data_full_duplex.txt** на **LXF DVD** или стр. 61 внизу) отображает данные ядра о сети при передаче файла между двумя компьютерами. Временная отметка 01:38:59AM (см. файл **nic_sar_data_half_duplex.txt** на **LXF DVD** или стр. 61 внизу) отображает данные ядра о сети при передаче файла между двумя компьютерами, при котором есть проблема с производительностью.

Обратите внимание на поле `coll/s` (конфликтов в секунду). Сетевой конфликт возникает, когда два (или более) устройства пытаются одновременно переслать данные по одному и тому же пути. Конфликт данных приводит к искажению сообщений, которые надо корректировать путем повторной отправки с хоста отправителя.

Что-то вызывает появление на хосте сетевых конфликтов, которые существенно замедляют работу сети. В идеальном случае у сетевой карты не должно быть конфликтов. Для получения данных сетевая карта на сервере была настроена на полудуплексный режим 100 Мбит/с вместо режима по умолчанию (100 Мбит/с, полнодуплексный режим). Неправильно настроенная сетевая карта влияет на производительность сети при передаче данных.

Закругляемся

Вот и все, наши верные читатели; урок по утилитам для исследования счетчиков ядра закончен. В ядре Linux есть внутренние счетчики, которые следят за запросами, временем выполнения, количеством блоков ввода/вывода и другой полезной информацией, которую просто нужно изучить и которая при правильном использовании поможет узнать многое о деталях работы системы. Одна из проблем со счетчиками — то, что перед их использованием нужно иметь довольно прочные знания о том, что они делают. Использование различных утилит для чтения значений счетчиков — отличный способ попрактиковаться, чтобы при возникновении реальной проблемы у вас уже был опыт. Помните, что если запуск утилит *sysstat* уже настроен в *cron*, то какие-то данные у вас уже есть. **LXF**

Ответ

Единственно «правильного» ответа на вопрос, который мы поставили в начале нашего урока, не существует, но некоторые ответы будут «правильнее» других. Проблемы с производительностью довольно-таки нередко бывает трудно решить. Чтобы сформулировать ответ, нужно разбить систему на четыре части: аппаратная часть, программы, сеть и пользователи.

Аппаратная часть включает все компоненты сервера, в том числе и массивы внешних жестких дисков. Программы — это операционная система, СУБД и приложения. К сети относятся все сетевые компоненты от хоста до сервера, а также среда, посредством которой осуществляется соединение.

Составив список из четырех частей, опишите, как они влияют на производительность. Диски,

если они работают медленно, повлияют на скорость чтения и записи. СУБД, не использующая индексов в часто опрашиваемом файле, тоже снизит производительность. Замена жестких дисков с целью их ускорить не решит проблему с индексами. Вы заметите улучшение производительности, но иногда по-прежнему будете получать жалобы на то, что система работает медленно.

Systemd: Нырнем в journalctl

Нейл Ботвик доказывает свои таланты журналиста, исследуя и используя множество функций в журнале *Systemd*.



Наш эксперт

У **Нейла Ботвика** огромный опыт работы с загрузкой, ведь у него по компьютеру в каждой комнате, а вот с опытом перезагрузки у него похуже, поскольку он работает в Linux.

Поскольку дистрибутивы массово переходят на *Systemd*, появляется множество новых способов работы, с которыми следует познакомиться. Один из них — журнал *Systemd*. Но что такое журнал? Чем он отличается от *syslog*? Как его использовать, и можно ли работать по-старому, если нам так больше нравится?

Журнал *Systemd* собирает и хранит данные, зарегистрированные в традиционном *syslog*, зарегистрированные сообщения ядра, стандартные сообщения и ошибки от системных сервисов и других источников. Журнал регистрирует больше метаданных по каждому сообщению, чем более старые демоны, и сохраняет все в структурированном, индексированном формате, который позволяет осуществлять быстрый и гибкий поиск. Основная команда для чтения из журнала — **journalctl**. Запустите ее без параметров, и вы увидите информацию, отображаемую в том же виде, что и в традиционном системном лог-файле. Однако есть несколько отличий: во-первых, **journalctl** автоматически использует при отображении журнала системный пейджер, обычно **less**. Вы также заметите, что он уходит назад вплоть до момента первичной установки. Причина в том, что журнал не использует **logrotate** для переключения с более старых данных, а вместо этого сжимает старые файлы журнала, но поиск осуществляет по всем этим файлам. Файлы журнала хранятся, пока их объем не превысит определенной величины, установленной в `/etc/systemd/journald.conf`.

Пусть он будет свеж

Хотя наличие длинной истории системных сообщений весьма удобно, вы, возможно, заметили некую задержку при запуске **journalctl**. К счастью, в **journalctl** есть способы сузить поиск, а поскольку журнал индексируется, это делается очень быстро. Например,

```
[nelz@hactar ~ 0]% journalctl --no-pager -b -p err
-- Logs begin at Fri 2015-02-06 19:34:39 GMT, end at Wed 2015-04-01 17:06:01 BST.
--
Apr 01 16:55:05 hactar systemd[1]: Failed to start Load Kernel Modules.
Apr 01 16:55:05 hactar systemd-modules-load[171]: Failed to find module 'vboxnetfl
t'
Apr 01 16:55:05 hactar systemd-modules-load[171]: Failed to find module 'vboxnetad
p'
Apr 01 16:55:30 hactar systemd-udev[408]: invalid key/value pair in file /lib64/u
dev/rules.d/97-hid2hci.rules on line 27, starting at character 244 ('/')
Apr 01 16:55:31 hactar systemd[592]: Failed at step EXEC spawning /usr/lib/systemd
/scripts/mdadm_env.sh: No such file or directory
Apr 01 16:57:01 hactar pulseaudio[1059]: [pulseaudio] bluez5-util.c: GetManagedObj
ects() failed: org.freedesktop.systemd1.LoadFailed: Unit dbus-org.bluez.service fa
iled to load: No such file or directory.
Apr 01 16:57:01 hactar pulseaudio[1208]: [pulseaudio] pid.c: Daemon already runnin
g.
Apr 01 16:57:01 hactar pulseaudio[1210]: [pulseaudio] pid.c: Daemon already runnin
g.
[nelz@hactar ~ 0]% █
```

» Это сообщение сплошь красного цвета, потому что опция `-p err` показывает только ошибки.

```
journalctl -b
```

показывает сообщения журнала с момента последней перезагрузки. Вы также можете использовать такие опции, как

```
journalctl --since today
```

```
journalctl --since 09:30 --until 11:00
```

Первая показывает записи после полуночи, вторая — за установленный промежуток времени. Вы можете использовать отметки даты и времени, а также строки типа **сегодня** и **вчера** в качестве аргументов для `--since` и `--until`.

Журнал *Systemd* хранит кучу данных, больше, чем стандартный *syslog*, так что вы, вероятно, будете ошарашены количеством информации, которую он вам выдаст. Большая часть информации, хранящейся в любом системном логге, является довольно малозначимыми данными, поэтому давайте искать действительно важное:

```
journalctl -b -p err
```

Отобразятся только сообщения о серьезных ошибках или еще более важные, с момента последней загрузки. Иногда вам просто нужны самые последние сообщения:

```
journalctl -n 10
```

```
journalctl -f
```

Первая команда показывает последние 10 сообщений, а `-f` следит за журналом, показывая сообщения по мере их добавления. Это похоже на использование **tail -f /var/log/messages**, но без буфера записи.

Фильтры

Пока что сообщения **journalctl** выглядели как стандартный файл `/var/log/messages`, но где же все эти дополнительные метаданные? Чтобы их увидеть, запустите

```
journalctl -o verbose
```

Вы видите, что здесь хранится большая часть дополнительной информации, и хорошая новость в том, что вы можете осуществлять поиск по журналу, используя ее. Больше нет необходимости в **grep** для лог-файла. Скажем, у вас есть проблемы с разделами *Samba*, которые появились только сегодня; вы можете найти соответствующие записи логга с помощью:

```
journalctl --since today _COMM=smbd
```

`_COMM` соответствует имени команды, и вы также можете использовать `_EXE` для соответствия полному пути. Фактически, **journalctl** старается по-умному работать с аргументами, которые вы ему передаете, поэтому если вы сделаете путь исполняемым, **journalctl** будет рассматривать его как фильтр для `_EXE`. Если вы дадите ему путь к узлу устройств, он покажет вам сообщения ядра по этому устройству, чтобы вы могли использовать такие команды:

```
journalctl /usr/sbin/crond
```

```
journalctl /dev/sda
```

Работаем по старинке

Есть несколько причин, по которым может понадобиться хранить традиционные текстовые лог-файлы в `/var/log`. Если вы запустите `logcheck` или `logwatch`, чтобы просканировать ваши системные логи на предмет подозрительных записей, надо представить им стандартный лог-файл, типа `/var/log/messages`. Простейший способ сделать это — установить стандартную программу записей в системном журнале, например, `syslog-ng`. С этим нет никаких проблем, система будет успешно делать

записи и в журнале, и в файле — мы сами так делаем именно по этой причине, хотя надеемся, что сможем обойтись одной программой и логом, когда более распространенными станут анализаторы логов для журнала.

Другая причина — удобство. Вы привыкаете к новому способу доступа к своим системным сообщениям, а вместе с ним работает и старая система. Насколько эта идея хороша — вопрос спорный: из-за возможности вернуться к `syslog-ng` для каждого

поиска по логам на освоение работы с `journalctl` у вас уйдет больше времени. Иногда лучше учиться плавать сразу на глубине!

По какой бы причине вы ни решили использовать другую программу регистрации системных сообщений, вам нужно будет установить также `logrotate`. В отличие от журнала `Systemd`, большинство демонов `syslog` не заботятся об удалении старых записей, а `logrotate` проследит, чтобы они не переполнили ваш жесткий диск.

```
Wed 2015-04-01 17:09:07.811877 BST [s-bc61e626af6c48308c269f208376e495;i-7a310;b-24dc5
TRANSPORT=stdout
PRIORITY=4
SYSLOG_IDENTIFIER=org.kde.kuiserver
UID=1000
GID=100
COMM=dbus-daemon
EXE=/usr/bin/dbus-daemon
CHDLINE=/usr/bin/dbus-daemon --fork --print-pid 5 --print-address 7 --session
CAP_EFFECTIVE=0
SYSTEMD_CGROUP=/user.slice/user-1000.slice/session-c1.scope
SYSTEMD_SESSION=c1
SYSTEMD_OWNER_UID=1000
SYSTEMD_UNIT=session-c1.scope
SYSTEMD_SLICE=user-1000.slice
MACHINE_ID=7d51f99e32d4b99f75519ebd0002372
HOSTNAME=hactar
BOOT_ID=24dc0311a34847f88b294aa5b5ea1b19
[1;39mMESSAGE=kuiserver(1002) JobView: pendingCallFinished: no more async calls$
PID=801
```

➤ Запустите опцию `-o verbose`, чтобы увидеть дополнительные метаданные, имеющиеся в каждой записи журнала. Все они пригодятся для поиска по журналу.

Есть разные области, которые вы можете охватить; они перечислены на map-странице `systemd.journal-fields`, но вы сможете и сами увидеть их, применив опцию `verbose`. Конечно, вы не захотите лезть в документацию всякий раз, когда вам нужно будет узнать, что искать, да вам и не придется этого делать. Вам поможет автодополнение `Bash` по клавише `Tab`: вы хотите задать поиск по `Systemd`, но не можете вспомнить имя поля? Попробуйте

```
journalctl _SYS<TAB>
```

Дополнение по `Tab` пригодится и для значений, хотя может работать довольно медленно, когда вы впервые будете осуществлять поиск по определенному полю. Можно использовать фильтр более чем для одного поля. Если вы укажете одно и то же поле дважды, например:

```
journalctl _SYSTEMD_UNIT=cronie.service _SYSTEMD_UNIT=collectd.service
```

невяно сработает OR [ИЛИ], подойдет любое совпадение. Если вы делаете фильтр по разным областям, `journalctl` применит к ним AND [И], и должны быть совпадения по обоим, если вы не разделите их с помощью `+`.

Настройки журнала

Вы уже видели, что `journalctl` по умолчанию передает свои сообщения на ваш пейджер. Есть два способа этого избежать. Вы можете использовать опцию `--no-pager` или направить сообщения не в терминал, а куда-нибудь еще, или перенаправить их в файл либо в другую программу. Для расширенных опций поиска могут быть случаи, когда вы решите запустить сообщения через `grep` или `awk`.

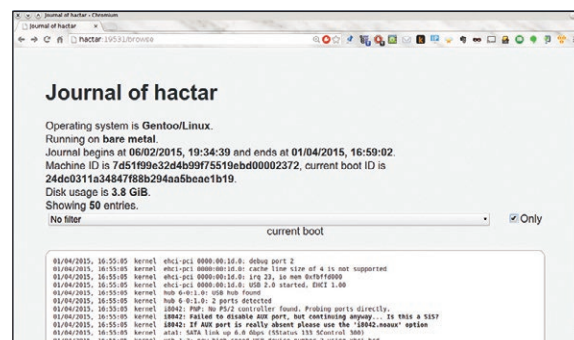
Журнал хранится в памяти для более быстрого доступа, но его можно записать и на диск. Если существует директория `/var/log/journal` и пользователь `systemd-journal` имеет право записи в нее, то ваши записи журнала будут храниться там. Поскольку журнал также хранится в памяти, `journald` регистрирует все с момента запуска `Systemd`, и затем переносит это на диск, когда директория `journal` становится доступной. Это означает, что он также хранит раннюю информацию по загрузке и сообщения

ядра, тогда как традиционные программы регистрации в системном журнале начинают вести регистрацию только после запуска сервиса `syslog`. Как уже упоминалось, `journald` также позаботится о перемещении и сжатии старых записей, но вместо использования метода «хранить x копий» `logrotate`, он использует место на диске для определения того, что сохранять. По умолчанию журналы ограничены использованием не более 10% общего размера файловой системы, и должны оставлять не менее 15% свободными. Если оба этих ограничения превышены, более старые журналы удаляются. Эти пределы можно установить, изменив `SystemMaxUse` и `SystemKeepFree` в `/etc/systemd/journald.conf`. Здесь можно настроить и ряд других опций, таких, как сжатие или передача сообщений в традиционный `syslog` (см. врезку «Работаем по старинке»), чтобы узнать, как и зачем вам может это понадобиться).

Удаленные логи

Конечно, вы можете зайти на удаленный компьютер с помощью `SSH`, чтобы использовать `journalctl`, но можете также включить сервис `systemd-journalgatewayd`, чтобы логи стали доступными через `HTTP` (на порте 19531 по умолчанию также доступно `HTTPS`). Хотя просмотр логов в браузере может показаться ограниченным, вы можете потом запустить `systemd-journal-remote`, чтобы перенести записи журнала из системы в отдельный журнал на локальном компьютере. Помните, `journalctl` накапливает записи со всех журналов, хранящихся на компьютере, так что показывает и локальные, и удаленные события. Вы можете сделать фильтрацию по `_HOSTNAME`, чтобы видеть записи от определенных систем.

В дополнение к системным журналам, `Systemd` также поддерживает журналы для зарегистрированных пользователей. Читать системный журнал могут только суперпользователь и члены групп `systemd-journal`, `adm` и `wheel`. По умолчанию `journalctl` показывает информацию из всех журналов, доступных активному на данный момент пользователю. Чтобы ограничить вывод системным журналом или журналом активного пользователя, используйте ключи `--system` или `--user`. LXF



➤ Просматривать логи в браузере — не идеально, но опция сообщений HTTP позволит вам также поделиться ими с другим компьютером для централизованного ведения журнала.

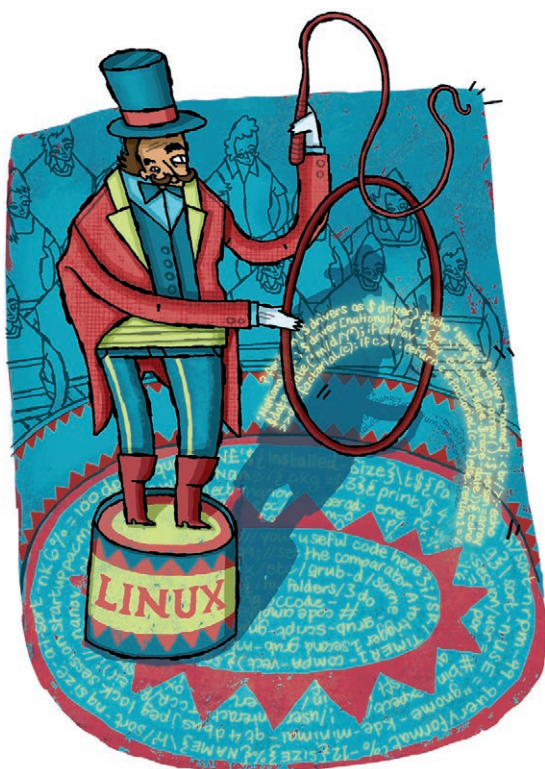
Sudo: Контроль root-доступа

Нейл Ботвик наслаждается божественным величием суперпользователя, исполняя мечты низших существ с помощью **sudo**.



Наш эксперт

У Неила Ботвика море опыта по части загрузки — ведь у него в каждой комнате по компьютеру; а вот в перезагрузке, после перехода с Windows на Linux, он не силен.



Администратор в системах Linux — на особом положении. Даже если вы единственный пользователь своего компьютера, для решения определённых задач всё равно понадобятся привилегии root, он же — суперпользователь. В некоторых дистрибутивах, например, в Ubuntu, данная

учетная запись скрыта — вы не можете войти в систему как root, но он всё же есть: без него работа системы невозможна.

Так почему же иногда приходится работать от имени root? Как это делается? И как передать это право другим пользователям, не давая им полную свободу действий в своей системе?

Традиционный способ получить доступ root — использовать команду **su**. Она позволяет переключиться на другого пользователя (не суперпользователя) и действовать от его лица на время выполнения команды. При запуске в оболочке без каких-либо аргументов **su** даёт доступ и к учётной записи суперпользователя (если вы знаете пароль). Это не совсем то же, что войти в систему как root: некоторые элементы текущего окружения всё же сохраняются. Максимально приблизиться к среде root можно, добавив опцию **-l**.

Как уже было сказано, **su** отвечает за смену пользователя, и по умолчанию это переход к root, но вы можете указать команде любое имя пользователя, например:

```
$ su -l fred
```

Запустив **su**, вы будете работать как другой пользователь, пока не введёте **exit** или не нажмёте **Ctrl+D**. Однако оставлять открытым сеанс с доступом root бывает небезопасно. Опция **-c** позволит передать привилегии только на время одной команды:

```
$ su -c "ifconfig eth0 up"
```

В этом случае, после ввода пароля, указанная команда выполняется без запуска сеанса терминала.

Тут пришел sudo

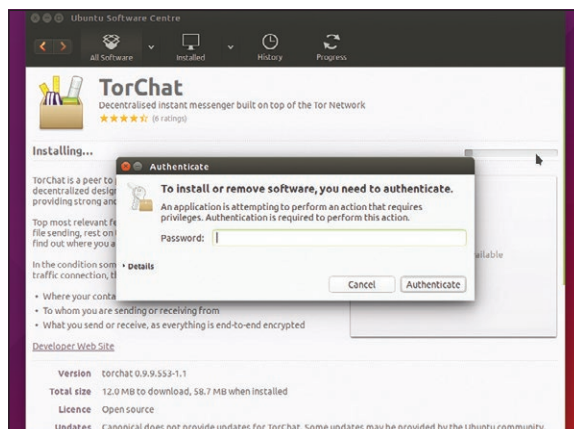
У традиционной **su** есть свои недостатки: пользователю надо сообщить пароль администратора, с которым он сможет вытворять всё что угодно, да ещё оставить открытым сеанс с доступом root. Метод **su -c** неудобен: ведь при выполнении нескольких команд вам придётся каждый раз вводить пароль. В таких случаях решением будет **sudo** — эта команда ныне используется в большинстве дистрибутивов.

Принцип работы **sudo** иной. Во-первых, **sudo** используется для выполнения одной команды, например:

```
$ sudo ifconfig eth0 up
```

Примечание: Теперь не надо заключать команду в кавычки! А во-вторых, она запрашивает ваш пароль, а не пароль суперпользователя, так что последний не надо разглашать (на самом деле, в Ubuntu его по умолчанию и нет). Запустив через **sudo** ещё одну команду, вы увидите ещё одно отличие: пароль не надо вводить снова. **Sudo** также позволяет без ввода пароля выполнить другую команду в течение пяти минут после предыдущей. При желании, это время ожидания можно изменить в файле настройки. Но если пользователь может запустить команду как root, просто используя свой пароль, что же помешает ему запустить вообще любую команду? Ответ помещается в файле настройки в **/etc/sudoers**.

» В различных дистрибутивах есть графические версии **sudo**, чтобы выполнять команды от имени root с рабочего стола.



Именно там подробно расписаны все права, и можно чётко контролировать все привилегии.

По умолчанию, любые команды разрешается выполнять членам группы `wheel`. В некоторых дистрибутивах эта группа может называться по-другому, например, `admin` (`wheel` — то есть как бы «рулевое колесо» — это традиционное название для группы с повышенными привилегиями, которое употреблялось ещё в Unix). Большинство инсталляторов включают в эту группу учётную запись, создаваемую при установке, а прочих пользователей в неё следует добавлять отдельно. Это означает, что пользователь, установивший дистрибутив, получает права администратора автоматически и ему не требуется не только доступ `root`, но даже пароль.

Visudo — это вам не vi

Вы можете передать привилегии `sudo` другим пользователям, отредактировав `/etc/sudoers`, но делать это вручную нельзя. Допустив синтаксическую ошибку, вы можете нарушить работу `sudo`. По соображениям безопасности, вам нужен доступ `root` даже для того, чтобы этот файл прочитать: ведь исправить ошибку редактирования будет невозможно.

Решением будет команда `visudo`, которая создает временную копию `/etc/sudoers` и загружает её в ваш редактор. Когда вы отредактируете файл, `visudo` проверит его правильность перед заменой реального `/etc/sudoers`. Судя по названию, для редактирования используется `vi`, но не бойтесь, это только запасной вариант. На самом деле, `visudo` использует редактор по умолчанию, указанный в переменной `EDITOR` рабочего окружения или заданный для системы в целом (в Ubuntu и других дистрибутивах на базе Debian).

Чтобы изменить редактор по умолчанию, введите

```
$ sudo update-alternatives --config editor
```

Настроить переменную `EDITOR` можно в профиле пользователя или посредством командной строки:

```
$ EDITOR="gedit" sudo -E visudo
```

Опция `-E` необходима, чтобы убедиться: переменная среды передается суперпользователю. Подробнее об этих переменных мы поговорим позже. Теперь мы знаем, как редактировать файл `sudoers` — давайте посмотрим, что же можно с ним делать.

Файл содержит два основных типа записей: алиасы (псевдонимы), т.е. определения переменных, и характеристики пользователей. Последние включают права. Например, в файле по умолчанию есть строка

```
%wheel ALL=(ALL) ALL
```

Базовый формат этой строки — «кто где = (в качестве_кого) что». Знак `%` в начале имени указывает, что это группа, и утверждение применимо ко всем её членам. «Где» — это имя хоста, с которого подается команда; следующие значения определяют, от имени каких пользователей можно запускать команду. Последним приводится список дозволенных команд. `ALL` — это универсальный символ `sudo`, то есть строка здесь обозначает, что члены группы `wheel` могут выполнять что угодно и где угодно; это стандартный вариант для пользователей, наделённых правами администратора.

Если вы хотите, чтобы такие привилегии были у другого пользователя, проще всего будет добавить его в группу `wheel` (или `admin`, в зависимости от вашего дистрибутива):

```
$ sudo gpasswd -a george wheel
```

Ограничение доступа к root

А если вы, например, хотите запретить пользователю `george` любые административные действия, кроме редактирования сетевых файлов, принадлежащих пользователю `Apache`?

Вам понадобится нечто вроде

```
george ALL=(apache) ALL
```

Вход от имени root

Хотя в некоторых дистрибутивах учетная запись `root` отключена, можно это обойти, с помощью одной из следующих команд:

```
$ sudo bash
```

```
$ sudo --login
```

Желая такое предотвратить, можете использовать `!`, запретив выполнение этих команд конкретному пользователю, например:

```
%wheel ALL=(ALL) ALL, !/bin/bash,!/bin/sh,!/bin/zsh
```

Зачем это нужно? В том числе потому, что каждая команда, выполненная с `sudo`, фиксируется в журналах, и в многопользовательской системе легко отследить, кто и что сделал. А при переключении в сеанс `root`, как описано выше, это становится невозможным.

Отсутствие `%` в начале означает, что это имя пользователя. Мы по-прежнему разрешаем ему выполнять все команды на всех хостах, но только от лица `Apache`.

```
$ sudo emacs index.html
```

вернёт ошибку, поскольку `sudo`, как и `su`, по умолчанию переключает на суперпользователя. Зато следующая команда выполнится, поскольку упоминается разрешённый пользователь:

```
$ sudo -u apache emacs index.html
```

Разумеется, `george` по-прежнему сможет выполнять любые команды, не требующие прав `root`, но зона их действия будет строго ограничена. Если мы хотим, чтобы он мог только редактировать файлы, то можем сделать так:

```
george ALL=(apache) /usr/bin/emacs, /usr/bin/vi
```

Вместо `ALL` в конце мы поставили список команд, указав полный путь к ним и разделив их запятыми. Можно также добавить допустимые аргументы:

```
george ALL=(ALL) /bin/mount /dev/sd[c-f]*, /bin/umount /dev/sd[c-f]*
```

Теперь `george` сможет создавать и удалять файловые системы — за исключением `sda` или `sdb`, то есть только на съёмных дисках. Ещё одним важным добавлением к этой строке (хотя в документации их намного больше), станет опция `NOPASSWD`, указывающая, что пользователю не будут отображаться подсказки для пароля — используйте её с осторожностью!

```
george ALL=(apache) NOPASSWD: /usr/bin/emacs, /usr/bin/vi
```

Переменные окружения

Чтобы отслеживать работу системы, Linux использует ряд переменных окружения, таких как `$HOME` и `$DISPLAY`. По умолчанию при выполнении команды `sudo` стирает большинство из них, хотя это и не всегда нужно. Самый грубый способ это обойти — запустить `sudo` с опцией `-E`, чтобы сохранить их все. Более продуманное решение — указать те, что вы хотите сохранить, в `sudoers`; например, вот это, возможно, потребуется вам при запуске через `sudo` программ с графическим интерфейсом:

```
Defaults env_keep += "DISPLAY HOME"
```

Существует множество других настроек, которые можно добавить в `/etc/sudoers`; все они подробно описаны в документации, но для начала этого вам будет более чем достаточно. **LXF**

```
net: sudo - Konsole
File Edit View Bookmarks Settings Help
[nelz@hactar /lxf dvd 0]% cd
[nelz@hactar ~ 0]% sudo visudo
Password:
I can't hear you -- I'm using the scrambler.
Password:
Have you considered trying to match wits with a rutabaga?
Password: █
```

► При вводе некорректного пароля конфигурация по умолчанию допускает мягкие угрозы по вашему адресу. Можете их отключить, для пущей солидности.

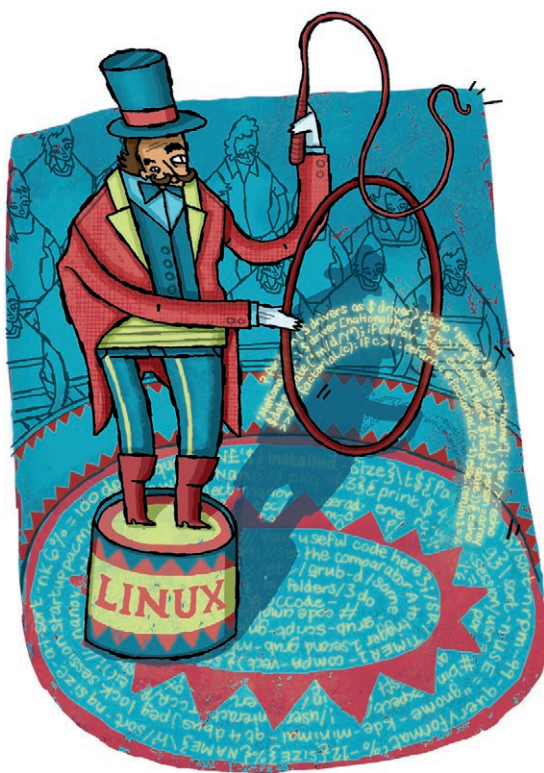
SystemTap: Мониторинг

Михалис Цукалос рассказывает обо всём необходимом, чтобы решиться взяться за SystemTap на Linux.



Наш эксперт

Михалис Цукалос — администратор UNIX, программист, администратор баз данных и математик; а ещё ему нравится писать статьи и узнавать что-то новое.



Эта статья о программе SystemTap, которая предоставляет простой интерфейс командой строки и язык скриптов с целью написания инструментария для работающего ядра, а также приложений, запущенных в пользовательской области. SystemTap — программа с расширенными возможностями, и важно понять, что для тестирования работающей системы с помощью SystemTap не надо перекомпилировать код, переустанавливать и исполняемую программу или перезагружать систему.

В Ubuntu SystemTap устанавливается командой `apt-get install systemtap systemtap-server systemtap-sdt-dev systemtap-grapher systemtapruntime` (от имени root). Версию SystemTap вам сообщит команда `stap --version`.

Также понадобятся установленные пакеты `-kbuild-<версия_ядра>` и `linux-image-<версия_ядра>-<architecture>-dbg` (оба имени зависят от версии вашего ядра) (рис. 1). Также необходимо включить в ядро информацию по отладке, чтобы SystemTap мог читать и отображать информацию. Иногда это означает, что вам понадобится установить ядро с отладочной информацией из исходника! В Ubuntu вы можете избежать процесса установки из исходника путём скачивания символического файла, соответствующего

вашему ядру, с <http://ddebs.ubuntu.com/pool/main/l/linux> и его установки (`# dpkg -i <ddeb FILE>`). Если вы не можете найти соответствующий символический файл, обновите свое ядро или скомпилируйте его самостоятельно. Не забудьте перезагрузить свою систему, чтобы изменения вступили в силу. Если вы по-прежнему сталкиваетесь с ошибками, примените команду `/usr/bin/stap-prep` для дополнительных подсказок.

Основная команда, которую вы будете использовать, называется `stap`. Попытку запустить `stap` от имени обычного пользователя ждёт провал. Следующее сообщение об ошибке проясняет ситуацию:

```
Вы пытаетесь запустить systemtap как обычный пользователь.
```

```
Вы должны быть root или членом группы stapusr, или, возможно, одной из групп stapusr или stapdev.
```

Поскольку работать от имени root — идея сомнительная, в указанные группы стоит добавить пользователей с правом выполнять `stap`. После исправления вышеупомянутой проблемы запуском `usermod -a -G stapdev,stapusr,stapusr mtsouk` от имени root настанет время запустить серверный процесс `systemtap`. При попытке запустить его от имени root SystemTap пожалуется и предостережёт вас от запуска серверного процесса, потому что «Вызов `stap-serverd` от имени root не поддерживается по соображениям безопасности». Так что запустите сервер, используя учетную запись другого пользователя. Поскольку сервер SystemTap пишет данные в две директории, перед успешным запуском серверного процесса ещё потребуется от имени root скомандовать

```
# chown -R mtsouk:mtsouk /var/run/stap-server
```

```
# chown -R mtsouk:mtsouk /var/log/stap-server
```

Если там нет этих двух директорий, создайте их от имени root, прежде чем менять разрешения. Теперь вы сможете запустить (`stap-server start`) и остановить (`stap-server stop`) `stap-server` без всяких предупреждений или сообщений об ошибках.

Так как вы получите доступ к ряду различных системных ресурсов, включая системные устройства, хорошей практикой будет запускать все команды SystemTap от имени root через команду `sudo`. Тем не менее, владельцем серверного процесса по-прежнему должен быть обычный пользователь. Если серверный процесс SystemTap уже запущен, показателем того, что SystemTap работает должным образом, будет следующий вывод:

```
$ sudo stap -v -e 'probe kernel.function("sys_open") {log("Hello World!") exit()}'
```

```
Pass 1: parsed user script and 95 library script(s) using 90552virt/27192res/2520shr/25500data kb, in 110usr/10sys/118real ms.
```

```
...
```

```
Pass 5: starting run.
```

```
Hello World!
```

```
Pass 5: run completed in 10usr/20sys/4630real ms.
```

Скорая помощь



Если вы администрируете только Linux, то просто обязаны изучить и SystemTap, и `perf_events`. Если же вы администрируете различные машины UNIX, лучшим выбором будет изучение DTace.

Как работает SystemTap

У *SystemTap* есть зонды для сбора информации и наборы *tapset* — группы связанных зондов. Событие и соответствующий ему обработчик вместе называются зондом. *SystemTap* может наблюдать на многих уровнях, из ядра, библиотек и приложений за транзакциями баз данных. Примерами *tapset* являются *syscall*, *ioblock*, *memory*, *scsi*, *networking*, *tcp* и *socket*.

Чтобы следить за всеми системными вызовами, введите `probe syscall.*`. Команда `stap -L 'syscall.*'`

отображает все возможные точки зондов, принадлежащих *tapset* `syscall`.

SystemTap сравнивает данный скрипт с существующей библиотекой *tapset* — обычно расположенной в `/usr/share/systemtap/tapset` — на предмет наличия использованных *tapset*. Затем *SystemTap* заменит любые обнаруженные *tapset* соответствующими им определениями в библиотеке. *SystemTap* переводит данный скрипт в C путём создания модуля ядра из него с помощью

компилятора C. Затем *SystemTap* загружает все модули и включает все зонды (события и обработчики событий) в скрипте с помощью утилиты `staprun`. Как только происходит какое-либо событие, выполняются соответствующие данному событию обработчики. Наконец, когда сессия *SystemTap* завершается, все связанные зонды отключаются и модуль ядра выгружается.

За полным списком *tapset* обратитесь к <https://sourceware.org/systemtap/tapsets>.

Строки в выводе, начинающиеся с **Pass 5:**, показывают, что *SystemTap* успешно создал инструментарий для проверки ядра, запустил инструментарий, определил событие, которое проверяется, и затем выполнил должный обработчик события.

Язык скриптов SystemTap

SystemTap поддерживает язык скриптов, похожий на AWK. Он состоит из двух основных компонентов: зондов [*probe*] и функций. Внутри них операторы и выражения используют C-образный оператор синтаксиса и приоритет.

Вернёмся к примеру “Hello World!”, который вы выполняли раньше. Событие `kernel.function("sys_open")` запускает обработчик события, помещённый в `{}`. Обработчик событий просто печатает сообщение “Hello World!” и выходит. Вместо `log()` могла бы быть использована более всесторонняя функция `printf()`. Функция `printf` *SystemTap* очень похожа на ее аналог в языке C.

Другой полезной функцией является `target()`; она используется для того, чтобы дать команду скрипту взять как аргумент ID процесса или команду. Это будет иметь тот же эффект, что и указание `if (pid() == process ID)` каждый раз, когда вы хотите выбрать определённый процесс. Использование `target()` делает ваши скрипты более всесторонними, гибкими и многократно используемыми, поскольку вы можете передать ID процесса в качестве аргумента каждый раз, когда вы захотите запустить ваши скрипты.

Несколько зондов

У скрипта *SystemTap* может быть несколько зондов для сбора информации, так что вы можете запускать команды двумя способами. Первый заключается в помещении кода *SystemTap* в командную строку, а второй — в использовании файла, содержащего код *SystemTap*, и исполнении его оттуда. Первый способ быстрее, однако второй лучше, особенно если надо выполнить несколько команд.

Команда `stap` запускает программу `staprun`, которая является движком для инструмента *SystemTap*. `Staprun` ожидает модуль ядра, произведённый внешней командой `stap`.

Следующий код наглядно демонстрирует различные части программы `stap`:

```
$ cat example.stp
probe begin { printf("Начали!\n") exit() }
probe * { printf("Вот здесь все и происходит!\n") exit() }
probe end { printf("Конец!\n") exit() }
$ sudo stap example.stp
Начали!
Вот здесь все и происходит!
Конец!
```

Как вы, вероятно, сообразили, часть `begin` всегда выполняется в первую очередь, а часть `end` — в последнюю. Часть `probe *` даёт команду *SystemTap* реагировать на каждое аппаратное устройство для сбора информации. Если у `probe *` нет условий выхода, его придётся завершить вручную, нажав Control+C.

Графическое отображение системы

Для получения высококачественного обзора того, что происходит в вашей системе, очень практичны гистограммы. *SystemTap* умеет генерировать гистограммы с помощью функции `@hist_log()`, которая выполняется после сбора данных. Следующая команда собирает данные для системного вызова `write()` и генерирует гистограмму степени двойки для возвращённых размеров записи:

```
$ sudo stap -ve 'global data; probe syscall.write.return { data <<<
$return; } probe end { printf("\ntrval (bytes)\n"); print(@ hist_
log(data)); }'
...
rval (bytes)
value |----- count
0 | 0
1 |@@@@@@@@@@@@ 46
2 |@@@@ 16
...
```

Для хранения полученных данных определяется глобальная переменная с именем `data`. По получении данных часть `end` используется для генерирования и печатания гистограммы.

Огненные диаграммы

Огненные, или пламенеющие, диаграммы [Flame graphs], созданные Брэнданом Греггом [Brendan Gregg], являются впечатляющим способом представления данных, собранных *SystemTap*. Огненные диаграммы отображают изучаемое ПО способом, »

```
2. mtsouk@mtsouk-VirtualBox: /tmp (ssh)
mtsouk@mtsouk-VirtualBox:~$ stap -e 'probe kernel.function("sys_open") {log("hello world") exit()}'
mtsouk@mtsouk-VirtualBox:~$ stap -e 'probe kernel.function("sys_open") {log("hello world") exit()}'
semantic error: while resolving probe point: identifier 'kernel' at <input>:1:7
source: probe kernel.function("sys_open") {log("hello world") exit()}
semantic error: missing x86_64 kernel/module debuginfo [man warning::debuginfo] under
'/lib/modules/3.13.0-46-generic/build'
Pass 2: analysis failed. [man error::pass2]
Tip: /usr/share/doc/systemtap/README.Debian should help you get started.
```

» Рис. 1. При отсутствии пакетов *SystemTap* не будет работать должным образом. Сообщения об ошибках помогут вскрыть суть проблемы и исправить её.



Вы можете узнать больше о *SystemTap* на <https://sourceware.org/systemtap>. “Systems Performance: Enterprise And The Cloud” Брэндона Грегга — отличная книга о производительности системы, тоже затрагивающая *SystemTap*. Вам также следует взглянуть в документацию на https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux.

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!

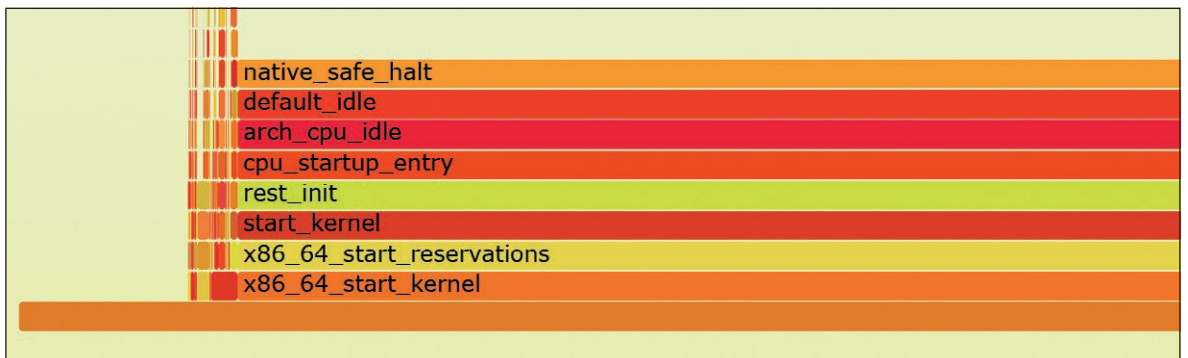
```

2. mtsouk@mtsouk-VirtualBox: ~/code/systemtap (ss)
systemtap:mtsouk$ sudo ./socket-trace.stp | head -30
 0 compiz(2109): -> sock_poll
84 compiz(2109): <- sock_poll
 0 compiz(2109): -> sock_poll
47 compiz(2109): <- sock_poll
 0 compiz(2109): -> sock_poll
29 compiz(2109): <- sock_poll
 0 compiz(2109): -> sock_poll
21 compiz(2109): <- sock_poll
 0 compiz(2109): -> sock_poll
15 compiz(2109): <- sock_poll
 0 compiz(2109): -> Sys_recvmsg
 5 compiz(2109): -> __sys_recvmsg
 9 compiz(2109): -> sockfd_lookup_light
11 compiz(2109): <- sockfd_lookup_light
15 compiz(2109): -> __sys_recvmsg
17 compiz(2109): -> copy_msghdr_from_user
19 compiz(2109): <- copy_msghdr_from_user
23 compiz(2109): -> sock_recvmsg
27 compiz(2109): <- sock_recvmsg
29 compiz(2109): <- __sys_recvmsg
31 compiz(2109): <- __sys_recvmsg
32 compiz(2109): <- Sys_recvmsg
 0 compiz(2109): -> Sys_recvmsg
 2 compiz(2109): -> __sys_recvmsg
 4 compiz(2109): -> sockfd_lookup_light
 5 compiz(2109): <- sockfd_lookup_light
 7 compiz(2109): -> __sys_recvmsg
 8 compiz(2109): -> copy_msghdr_from_user
10 compiz(2109): <- copy_msghdr_from_user
11 compiz(2109): -> sock_recvmsg
ERROR: Couldn't write to output 1 for cpu 0, exiting.: Success
systemtap:mtsouk$ cat socket-trace.stp
#!/usr/bin/env stap

probe kernel.function("@net/socket.c").call {
    printf ("%s -> %s\n", thread_indent(1), pfunc())
}
probe kernel.function("@net/socket.c").return {
    printf ("%s <- %s\n", thread_indent(-1), pfunc())
}
systemtap:mtsouk$
    
```

➤ Рис. 2. Запуск скрипта по имени `socket-trace.stp` в системе Linux генерирует вывод, похожий на представленный на этом рисунке. Скрипт очень удобен для понимания взаимодействия всех процессов с сетью на уровне ядра.

позволяющим легко определить наиболее общие пути кода. Используя огненную диаграмму, администратор может визуально отследить, какие функции вызываются другими функциями. Разные строки показывают, какие функции отдельной программы забирают чересчур большую часть времени CPU. Подобные



➤ Рис. 3. Огненная диаграмма — хороший способ визуализации данных, полученных от SystemTap.

отображения легко позволяют создать полную картину происходящего. Чтобы сгенерировать огненную диаграмму, понадобятся две внешние программы, но весь процесс относительно прост. Следующие команды иллюстрируют процесс:

```

$ sudo stap -s 32 -D MAXBACKTRACE=100 -D
MAXSTRINGLEN=4096 -D MAXMAPENTRIES=10240 -D
MAXACTION=10000 -D STP_OVERLOAD_
THRESHOLD=5000000000 --all-modules -ve 'global s; probe
timer.profile { s[backtrace()] <<< 1; } probe end { foreach (i in s+) {
print_stack(i); printf("\t%d\n", @count(s[i])); } } probe timer.s(60) {
exit(); }' > LXF.stap
...
$ head -5 LXF.stap
0xffffffff811e1c54 : mangle_path+0xc4/0xd0 [kernel]
...
0xffffffff81221dd3 : show_pid_map+0x13/0x20 [kernel]
$ ./stackcollapse-stap.pl LXF.stap > LXF.folded
$ cat LXF.folded | ./flamegraph.pl > LXF.svg
    
```

Важными опциями, необходимыми для правильного генерирования огненной диаграммы, являются `-D MAXBACKTRACE=100` и `-D MAXSTRINGLEN=4096`, поскольку они не позволяют обрезать строки в журнале трассировки стека. При желании можете поэкспериментировать с другими параметрами.

Пожалуйста, учтите, что предыдущая команда `stap` осуществит выход автоматически, поэтому вам не придется останавливать её, нажав `Control+C`. Собирайте требуемые вам данные, обрабатывайте их и генерируйте огненную диаграмму в виде SVG-файла. Оба скрипта можно добыть на <https://github.com/brendangregg/FlameGraph>.

Рис. 3 внизу изображает сгенерированную огненную диаграмму. Генерирование огненной диаграммы было простым; сложно правильно интерпретировать результаты, что зависит от вашего опыта и хорошего понимания наблюдаемой системы Linux.

Вы также можете использовать огненные диаграммы для визуализации данных, созданных другими инструментами контроля производительности, включая *DTrace*, *Windows XPerf*, *perf_events*, *OS X Instruments* и *Google Chrome Developer Tools*.

Скрипты для сокетов

Хорошая новость в том, что вы можете найти полезные и готовые к запуску скрипты на сайте *SystemTap*. Например, скрипт `sockettrace.stp` отслеживает функции, вызванные из файла ядра `net/socket.c`. Рис. 2 слева показывает небольшую часть вывода скрипта, запущенного на Ubuntu. Вывод довольно низкоуровневый; было бы крайне трудно получить подобную информацию иным способом, особенно в приложениях, где у вас нет прямого доступа к их исходному коду.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Полезные опции командной строки stap

SystemTap поддерживает множество опций и параметров командной строки. Перечислим наиболее полезные; для полного списка параметров ознакомьтесь с тап-страницей **stap**.

Опция **-s**, за которой следует число, даёт *SystemTap* команду использовать буферы с указанным числом мегабайт для передачи данных от ядра к пользователю. Если у вас работает мультипроцессор в режиме Bulk, то это число указано для каждого процессора. Опция **-d**, за которой следуют имя и значение, велит *SystemTap* добавить данную директиву препроцессора C модулю **Makefile**. Она часто применяется для переопределения предельных

параметров. Опция **-d** с последующим именем модуля даёт команду *SystemTap* добавить символическую/unwind информацию для данного модуля в модуль объекта ядра. Опция **--all-modules** эквивалентна указанию **-dkernel** и **-d** для каждого загруженного в данный момент модуля ядра. Эта опция может значительно увеличить модули зондов. Опция **-e** даёт команду *SystemTap* ожидать и запустить скрипт, который будет дан в командной строке. Если вы хотите заставить команду **stap** читать скрипт *SystemTap* из файла, укажите имя файла в командной строке безо всяких параметров. С опцией **-o** и последующим именем файла *SystemTap* отправит

стандартный вывод в файл с указанным именем. Опция **-v** велит *SystemTap* добавить объём предоставленной информации для всех путей. С опцией **-c** и последующей командой, указанной со своим полным путём, *SystemTap* настроит функцию обработчика событий **target()** на указанную команду. Аналогично, с опцией **-x** и последующим ID процесса *SystemTap* настроит функцию обработчика событий **target()** на указанный ID процесса.

Теперь должно быть ясно, что следующая команда **stap** аналогична команде **strace**:

```
$ sudo stap -c /bin/ls -ve 'probe syscall.* {printf("PID: %d\tNAME: %s\tARGSTR: %s\n", pid(), name, argstr); }
```

Скрипт **iotime.stp** отслеживает каждый случай, когда системный вызов открывает, закрывает, читает и пишет в файл. Вы можете остановить его вручную, нажав Control+C. Действительно полезно в этом скрипте то, что он печатает имя процесса, вызвавшего операцию с файлом, ID этого процесса и имя файла, к которому был получен доступ. Если процесс мог успешно написать или прочитать файл, в выводе появится пара строк **access** и **iotime** вместе. Представленный материал тоже довольно низкоуровневый; вы можете отфильтровать и изменить вывод, используя команду **grep**. Рис. 4 показывает небольшой участок вывода скрипта **iotime.stp**.

Вы также можете наблюдать за отдельными файлами с помощью скрипта **inodewatch.stp**. Вначале понадобится обнаружить низкоуровневую информацию командой **stat**. Следующий вывод иллюстрирует процесс мониторинга **/var/log/wtmp**:

```
$ stat -c '%D %i' /var/log/wtmp
801 813532
$ sudo ./inodewatch.stp 0x8 0x01 813532
accounts-daemon(1574) vfs_read 0x800001/813532
last(11805) vfs_read 0x800001/813532
```

Команда **stat** возвращает информацию, необходимую для запуска **inodewatch.stp**: **813532** — это номер инода файла, за которым вы хотите наблюдать, **8** — номер главного устройства, а **01** — номер второстепенного устройства файловой системы, где находится файл. **0x** перед номерами обозначает, что величины записаны в шестнадцатеричном формате.

Скрипты для процессов

Скрипт **topsys.stp** отображает топ-20 системных вызовов, использованных системой за 5-секундный интервал, а также то, сколько раз каждый системный вызов использовался в течение этого периода. Вы можете изменить интервал в строке кода **probe timer.s(5)**, или изменить число печатаемых системных вызовов в строке **foreach (syscall in syscalls_count - limit 20)**.

Скрипт **syscalls_by_proc.stp** отображает топ-20 процессов, совершающих наибольшее количество системных вызовов. Он также показывает, сколько системных вызовов совершил каждый процесс за определённый период. Его вывод должен выглядеть примерно так:

```
#SysCalls Process Name
20352 Xorg
3015 compiz
768 mysqld
...
```

SystemTap также работает на уровне сети. Скрипт **net.stp** отображает сетевую информацию в интерфейсе для каждой сети и на уровне имени для каждого процесса. Другие интересные

```
2. mtsouk@mtsouk-VirtualBox: ~/code/systemtap (ssh)
systemtap:mtsouk$ sudo ./iotime.stp | head -40
39770 1187 (vminfo) access /var/run/utmp read: 4608 write: 0
39774 1187 (vminfo) iotime /var/run/utmp time: 15
264265 11285 (df) access /etc/ld.so.cache read: 0 write: 0
264330 11285 (df) access /lib/x86_64-linux-gnu/libc.so.6 read: 832
264333 11285 (df) iotime /lib/x86_64-linux-gnu/libc.so.6 time: 2
264616 11285 (df) access /usr/lib/locale/locale-archive read: 0
264705 11285 (df) access /usr/share/locale/locale.alias read: 2
264707 11285 (df) iotime /usr/share/locale/locale.alias time: 6
264802 11285 (df) access /usr/share/locale-langpack/en/LC_MESSAGES
0 write: 0
264888 11285 (df) access /etc/mtab read: 812 write: 0
264890 11285 (df) iotime /etc/mtab time: 5
265403 11285 (df) access /usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache
write: 0
3925642 11286 (ls) access /etc/ld.so.cache read: 0 write: 0
3925708 11286 (ls) access /lib/x86_64-linux-gnu/libselinux.so.1 read: 832 write:
3925711 11286 (ls) iotime /lib/x86_64-linux-gnu/libselinux.so.1 time: 3
3925756 11286 (ls) access /lib/x86_64-linux-gnu/libacl.so.1 read: 832 write: 0
3925758 11286 (ls) iotime /lib/x86_64-linux-gnu/libacl.so.1 time: 2
3925806 11286 (ls) access /lib/x86_64-linux-gnu/libc.so.6 read: 832 write: 0
3925808 11286 (ls) iotime /lib/x86_64-linux-gnu/libc.so.6 time: 2
3925852 11286 (ls) access /lib/x86_64-linux-gnu/libpcre.so.3 read: 832 write: 0
3925854 11286 (ls) iotime /lib/x86_64-linux-gnu/libpcre.so.3 time: 1
3925893 11286 (ls) access /lib/x86_64-linux-gnu/libacl.so.2 read: 832 write: 0
3925895 11286 (ls) iotime /lib/x86_64-linux-gnu/libacl.so.2 time: 2
3925933 11286 (ls) access /lib/x86_64-linux-gnu/libattr.so.1 read: 832 write: 0
3925935 11286 (ls) iotime /lib/x86_64-linux-gnu/libattr.so.1 time: 1
3926319 11286 (ls) access /proc/filesystems read: 329 write: 0
3926321 11286 (ls) iotime /proc/filesystems time: 19
3926378 11286 (ls) access /usr/lib/locale/locale-archive read: 0 write: 0
3926493 11286 (ls) access /usr/share/locale/locale.alias read: 2570 write: 0
3926495 11286 (ls) iotime /usr/share/locale/locale.alias time: 6
3926604 11286 (ls) access /usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache read
write: 0
3926826 11286 (ls) access /etc/nsswitch.conf read: 507 write: 0
3926828 11286 (ls) iotime /etc/nsswitch.conf time: 6
3926855 11286 (ls) access /etc/ld.so.cache read: 0 write: 0
3926906 11286 (ls) access /lib/x86_64-linux-gnu/libnss_compat.so.2 read: 832 write:
3926908 11286 (ls) iotime /lib/x86_64-linux-gnu/libnss_compat.so.2 time: 1
3926953 11286 (ls) access /lib/x86_64-linux-gnu/libnsl.so.1 read: 832 write: 0
3926955 11286 (ls) iotime /lib/x86_64-linux-gnu/libnsl.so.1 time: 2
```

Через месяц:
Углубимся
в perf_event

скрипты, с которыми стоит познакомиться — **iosstats.stp**, **timeout.stp** и **stopwatches.stp**. Все вышеупомянутые скрипты *SystemTap* могут быть полезны для каждого системного или сетевого администратора, в зависимости от возникшей проблемы. Самый простой способ создать собственный скрипт — взять готовый скрипт и изменить его. Установив пакет **systemtap-doc**, вы автоматически получите большую часть скриптов, установленных в **/usr/share/doc/systemtapdoc/examples**.

Главная проблема с инструментами вроде *SystemTap* в том, что вам нужно регулярно использовать их, чтобы освоить, потому что «готовь сани летом, а телегу — зимой». Другими словами, не пытайтесь изучать *SystemTap*, когда проблема уже налицо!

Прочитав эту статью, вам также следует ознакомиться с учебниками в **LXF194** для введения в **DTrace**. **LXF**

► Рис. 4. Скрипт **iotime.stp**, запущенный на системе Linux. Скрипт полезен для отслеживания каждого открытия, закрытия, чтения и записи в файл системным вызовом.

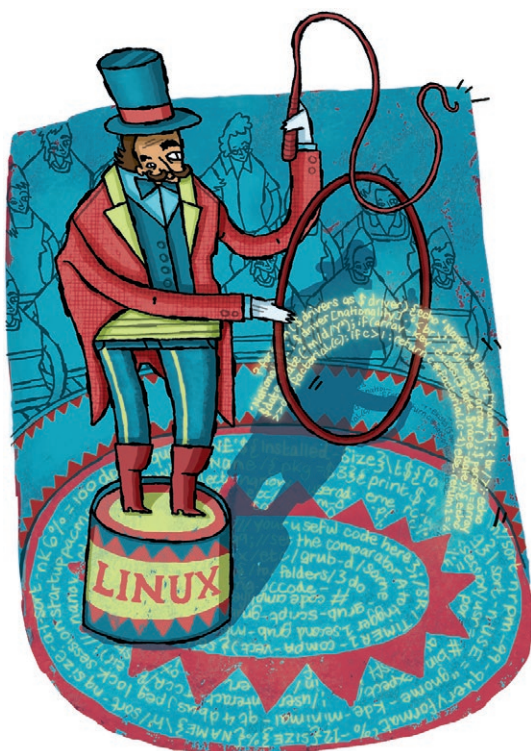
DD-WRT: Хакнем свой Wi-Fi-роутер

Мэтт Билби кратко описывает, как подкрепить центральное устройство вашей домашней сети собственным ПО.



Наш эксперт

Мэтт Билби устанавливал нестандартные прошивки на роутеры с тех времён, когда скорость Wi-Fi измерялась в почтовых голубях.



Установка нестандартной прошивки — почти всегда прекрасный опыт познания, хотя случается познать и чувство погубителя вполне исправного устройства. Вам, правда, может показаться, что в поломке нет вашей вины, но ваше желание продолжать влечёт понимание, что виноваты всё-таки вы, поскольку вы-то и потрошили роутер.

Прояснив это, продолжим: целесообразней всего использовать старый запасной роутер. Взглянем на это так: в итоге вы останетесь без гарантии производителя — так лучше уж и начинать без неё. К тому же менее вероятно испытать гложущее изнутри чувство вины, если во время обновления прошивки вы чихнёте и шнур питания отскочит; зато более вероятно раскрепостить новые функции. С другой стороны, столь нестандартная прошивка, как DD-WRT, способна отнять достаточно много времени для адаптации к новой технологии (и ещё больше — чтобы заставить её работать надёжно), так что с супер-роутером года можно сесть в лужу, даже если у вас хватило отваги к нему подступиться.

Поддержка роутера

Сначала — плохие новости. Без заметных исключений, комбинации роутер-модем работать не будут: например, знаменитая линейка «BT Home Hub» не поддерживается [BT — британская телекоммуникационная компания, — прим. пер.]. Но не всё потеряно, если вы подключены к оптоволоконной линии VDSL/BT, поскольку тогда можно использовать отдельный модем OpenReach и подключить роутер по вашему выбору. У других провайдеров может даже быть режим только для модема, который позволяет подключить свой собственный роутер: например, «Virgin Media's Super Hubs» попадают в эту категорию.

Если у вас отдельный роутер, не обязательно бросаться загружать на него новую прошивку. У одних роутеров неподходящий чипсет, у других не хватает флэш-памяти, а у третьих нет ОЗУ. Четвертые, честно говоря, дохловаты. Однако поддерживается на диво широкий диапазон роутеров. Как же узнать, что ваш — один из них?

Сначала зайдём на базу данных роутеров DD-WRT (www.dd-wrt.com/site/support/router-database). Просто введите номер своей модели в поле поиска и перекреститесь. База данных обычно прямо отвечает: да или нет, но не пляшите от радости, увидев свою модель в списке, пока не убедитесь, что совпадает и аппаратная версия — некоторые производители почти полностью меняют внутренности между версиями одной модели роутера.

Ради интереса попробуйте поискать в базе данных WRT54G и посчитайте количество вхождений. WRT54G — это дедушка для DD-WRT, и у него долгая история. Но учтите, что как минимум одна версия не поддерживается вообще, и спецификации могут сильно отличаться друг от друга. У многих, например,

Скорая помощь

Есть и другие прошивки, способные обоснованно соперничать за ваше внимание. В частности, различные ветки проекта Tomato, AsusWRT-Merlin (промежуточная между нестандартной и заводской прошивкой только для роутеров Asus) и даже OpenWRT — обширная база, на которой созданы многие другие.

В наше время приличный роутер сам, не приставая к хозяевам, делает свое дело, создавая раздолье для создания домашних сетей. Но почему бы не рискнуть сделать что-то по-своему? Если вы готовы к переменам, мир нестандартных прошивок поразит вас выбором конфигураций и заманчивым перечнем новых функций.

Выбрав прошивку DD-WRT, мы сумеем заставить эти аккуратные и неприязательные встраиваемые устройства отдуваться на максимум возможностей. Будет пот, а может, и слёзы, но мы проведём вас через процесс выбора и установки прошивки, покажем ряд элегантнейших способов ее надурить и откроем путь к новым дерзким начинаниям.

DD-WRT — лишь одна из многих нестандартных прошивок для беспроводных роутеров, но она является их душой, с широким кругом поддержки, относительной простотой использования, адекватным развитием и кладезем функций. Установка DD-WRT — отнюдь не мелкое дельце: она полностью изменит работу роутера и может открыть такую функциональность, как SSH, работа с файлами и мультимедиа, QoS, сети VLAN и VPN — на любой вкус, больше, чем укладывается в голову. Однако риск соизмерим с масштабом перемен.

сильно урезана флэш-память, и функции, основанные на ней, будут ограничены.

Для обнаруживших, что их роутер поддерживается, есть два основных светоча во мраке: вики DD-WRT и форумы сообщества. Вики отлично подходит для получения базового понимания любых проблем, способных повлиять на конкретный роутер. Начните со страницы поддержки устройств (www.dd-wrt.com/wiki/index.php/Supported_Devices). Ссылки с этой страницы нередко указывают, что ваш роутер имеет особое руководство по установке — возможно, просто ввиду популярности данной модели; но это может и означать, что процесс прошивки идёт с оговорками или спецификациями, так что имейте это в виду.

Надёжный форум друзей

Форумы — лучшее место, чтобы узнать, что работает прямо сейчас у других на таком же оборудовании (см. www.dd-wrt.com/phpBB2). Обратите особое внимание на темы, где обсуждается отказоустойчивость популярных или наиболее стабильных блогов. Следите за сообщениями «гуру», часто публикующих большие детальные списки роутеров и прошивок, с которыми они работают. Эти ребята выполнили своё домашнее задание, и вы тоже постарайтесь выполнить своё, даже если это иногда означает списывание с воображаемого соседнего стола.

DD-WRT существует в хронической версии бета, и самый новый релиз не всегда будет лучшим для вашего конкретного устройства. Нет ничего постыдного или унижающего в использовании сборки, значительно отстающей от последних достижений. Подходит для вашей аппаратуры — ну и берите её. Со старыми релизами главное — убедиться в отсутствии критических уязвимостей для себя и своей аппаратуры. Для начала, сборки с 19163 по 23882 — антивариант неудачный: компоненты, использующие OpenSSL, будут подвержены уязвимости HeartBleed. На ваше счастье, ни одна обычная сборка не затронута специфичной для *Bash* уязвимостью Shellshock; как и многие прошивки встроенных устройств, DD-WRT для обзаведения A Shell полагается на *BusyBox*.

Аналогично, использование *uclibc* означает, что Ghost, уязвимость *glibc*, сегодня уже не волнует. Однако использование нестандартной прошивки может потом резко ударить мячиком

«Форумы — лучшее место, чтобы узнать, что работает сейчас.»



➤ Марку или модель пишут обычно на наклейке на задней или нижней стенке роутера. Запишите версию в дополнение к номеру модели

безопасности на вашей стороне корта, так что вам действительно нужно быть в курсе любых новых уязвимостей.

Теперь рассмотрим практический пример. Возьмём роутер Cisco Linksys E3000, имеющий неплохой баланс возраста и пригодности. Ему около пяти лет, и у него нет новомодных технологий типа Wireless AC, но в своё время он был лидером, поддерживая одновременно диапазоны 2,4 и 5 ГГц. База данных роутеров показывает твердое «да», а на вики есть по нему некоторая конкретная информация. Особо отмечено наличие 60К NVRAM и требование прошивать промежуточную сборку (см. врезку «Промежуточные сборки и TFTP», стр. 74). Нужно принять во внимание обе особенности.

Нам повезло, как это иногда бывает: на форумах сборка от февраля 2015 г. (build 26138) преподносится как стабильная для Linksys серии E. Есть там какие-то дебаты об ошибке в реализации гостевой Wi-Fi, но это не стоит нашего времени.

Основной рассадник новых релизов DD-WRT находится на [ftp://ftp.dd-wrt.com/betas](http://ftp.dd-wrt.com/betas), и мы знаем из вики, что E3000-совместимые сборки можно найти в подпапке **broadcom_K26**. Можно без проблем выбрать там промежуточную мини-сборку для E3000, но если потом мы захотим перейти к большей общей сборке, то надо помнить о пределе в 60К NVRAM и выбрать сборку в 60К из той же папки. Огромная 60К-сборка (ужас!) слишком велика для нашего 8-МБ флэш-накопителя — хорошо, что мы вовремя проверили, не то дошло бы до подсчёта байтов; итак, мы начнём всё-таки с так называемой «большой сборки».

Время обновления прошивки

Пришло время проверить и перепроверить все наши источники информации, поскольку мы готовы обновить прошивку. Следующие шаги обычно применимы, но надо почитать о вашей модели: нет ли каких-либо различий.

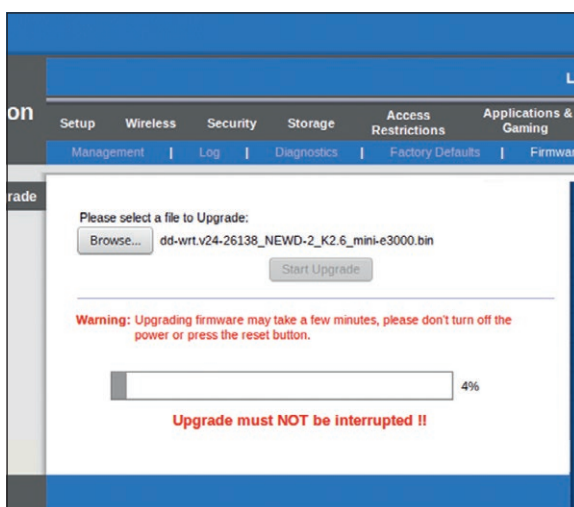
Во-первых, надо подключить компьютер к роутеру с помощью кабеля, затем настроить ему статический IP-адрес в той же подсети, что и роутер. Нельзя гарантировать, что всё будет плохо, если этого не сделать, но вы действительно хотите оставить роутер при деле в процессе промывания ему мозгов? Однозначно — нет. Не вы.

Скорая помощь

DD-WRT даст вам контроль, но не обязательно производительность. Если вас интересует только максимальная скорость, то прошивка производителя часто быстрее альтернативной.

Внимание

Следуя этому уроку, можно уничтожить свое оборудование. *Linux Format* не несёт ответственности (в т. ч. по неосторожности) за любые возможные повреждения, потерю данных или ущерб в итоге урока. Страх и риск — ваши.



➤ Сейчас самое время спокойно поразмыслить...

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!

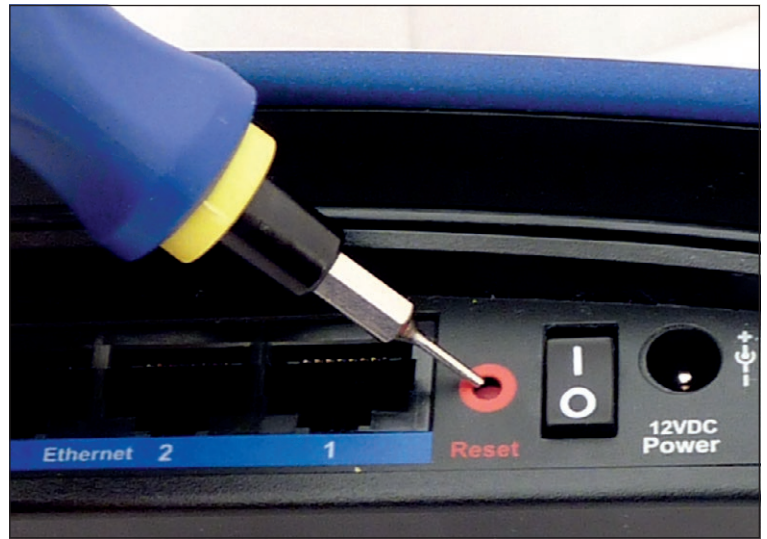
Сброс 30-30-30

Не стоит недооценивать, насколько криво всё будет, если временные старой прошивки не сработают с новой. Полный сброс 30-30-30 очистит NVRAM и вернёт большинство роутеров к прошивке по умолчанию, что надо делать до и после установки каждой новой версии прошивки.

Кнопка Reset находится, вероятно, на задней стенке устройства, иногда в углублении. Возьмите, если нужно, скрепку и примите удобное положение; кнопку сброса надо держать нажатой в течение 90 секунд или более. Это очень долго, и кого-то может свести судорогой.

Нажмите и удерживайте кнопку сброса в течение 30 с. Не отпуская её, вытащите сетевой штекер на задней стенке роутера. Отсчитайте ещё 30 с. Продолжая удерживать кнопку сброса, снова подключите роутер. Снова отсчитайте 30 с. Наконец, отпустите кнопку и встряхните руки, чтобы восстановить в них кровообращение. Роутер должен вернуться к значениям по умолчанию для установленной прошивки. (Руки можете опустить.)

Это для старых роутеров, а всё большее число новых AC-роутеров выполняют полный сброс другими способами. Если 30-30-30 не возвращает значения по умолчанию, проверьте, что работает для вашего роутера, и используйте этот метод.



► Можно, конечно, взамен скрепки приобрести инструмент.

вопросом, но, по чести, это иногда довольно полезно и стоит принятого риска.

DD-WRT радостно поддерживает удалённый доступ к графическому интерфейсу по HTTP или HTTPS. Вы ни в жизнь не захотите выдать миру доступ к сердцу вашей домашней сети без единого уровня безопасности, но можно подумать о разрешении HTTPS-соединения.

Хотя, постойте. Вот изящный трюк: почему бы не запретить удалённый доступ к web-интерфейсу вовсе, а подключаться по SSH? Вы можете войти и управлять роутером удалённо с помощью командной строки или создать SSH-туннель, дающий вам, по сути, локальный доступ к web-интерфейсу. Это будет работать из любого места, и надо только открыть одну дверь, чтобы позволить оба типа доступа. Давайте посмотрим, как это делается.

Во-первых, служба удалённого доступа по SSH включается в другой части web-интерфейса DD-WRT. Раздел Administration, вкладка Management. Здесь секция удалённого доступа. Не разрешайте удалённый Web GUI Management. Вместо этого разрешите SSH Management. Появится возможность выбрать для него порт. Незачем — и отнюдь не следует — использовать стандартный для SSH порт 22, в нашем примере мы возьмём порт 19198. Вы можете выбрать и другой, не беспокойтесь: соединение на этом порту пойдёт через службу SSH на роутере без дополнительных усилий с вашей стороны.

Теперь вы в состоянии подключиться через SSH к роутеру из внешнего мира точно так же, как из вашей локальной сети — единственное отличие заключается в том, что надо указать порт и использовать наружный IP вместо локального:

```
ssh -p 19198 root@WANIP
```

Замените WANIP глобальным адресом вашей локальной сети. Это может быть имя DNS или IP-адрес. Весьма вероятно, что ваш провайдер не предоставляет вам статический IP-адрес, но не обязательно следить за каждым его изменением. DD-WRT поддерживает автоматическое обновление ряда различных динамических

DNS-сервисов — взгляните на DDNS на вкладке Setup для различных вариантов.

Мы неплохо продвинулись, но как насчёт web-интерфейса? Ну, попробуйте начать сессию SSH такой командой:

```
ssh -p 19198 root@WANIP -L 8080:localhost:80
```

При этом, как и раньше, запустится сеанс SSH, но последняя часть команды создаёт туннель с порта 8080 на локальной машине на порт 80 роутера. Теперь попробуйте открыть в браузере следующий URL: <http://localhost:8080/>.

Ух ты. Круто! Вот оно. У вас есть свой web-интерфейс с удалённого доступа, и всё это шифруется через сессию SSH. Сейчас мир буквально в вашем распоряжении.

Бросаем вызов

Имея доступ через web-интерфейс и SSH, что ещё стоит попробовать? Да всё новое попробовать стоит! Считайте это вызовом.

Как насчёт создания SSH-туннеля, о чём мы говорили, чтобы запустить на своём домашнем роутере прокси SOCKS5, через который можно зашифровать ваш трафик, когда вы находитесь далеко от дома? Если у вас есть учетная запись VPN, не подключить ли как клиент свой роутер? (Это отлично подходит для подключения малодоступных для настройки встроенных устройств без поддержки VPN). Может, у вас есть широкополосный мобильный USB-модем? DD-WRT сумеет поиграть и с ним. Почему бы не попробовать подключить через роутер ещё одного интернет-провайдера, на случай, если ваш основной провайдер сгинет?

А если вы действительно хотите поиграть с огнём, можно организовать облачное хранилище на USB-диске, подключённом к роутеру. Вы ведь не планировали дать своему роутеру отставку?

Вот как-то так. Некоторые совершенно поразительные возможности, которые раньше казались волшебством, уже работают на устройстве, которое вы, наверное, уже забросили было на полку. Помните, что маршрутизация сетевого трафика для этого устройства — хлеб с маслом, и не бойтесь заставить его зарабатывать на жизнь! **LXF**

«Все новое попробовать стоит! Считайте это вызовом.»

Система: inotify и getopts

В третьей части серии д-ра Криса Брауна мы научимся отслеживать изменения в файловой системе, обрабатывать аргументы командной строки и прочее.



Наш эксперт

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

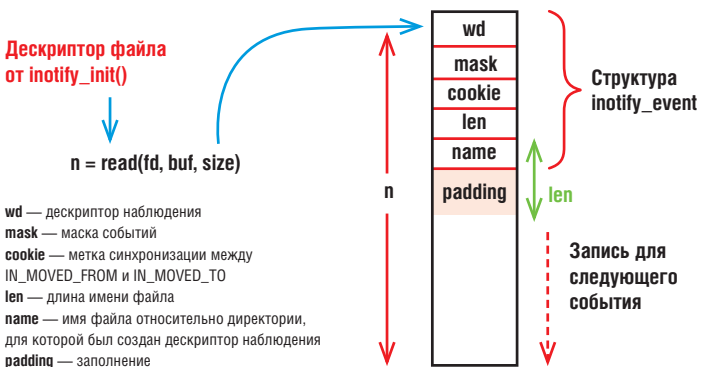
Представьте себе такую ситуацию: вы просматриваете содержимое каталога в графическом файловом менеджере. Вы добавляете в каталог новый файл — например, копируете его с командной строки; и — вуаля! — он мигом появляется в окне файлового менеджера. Откуда файловый менеджер узнал о появлении файла? Ну, он мог, например, следить за изменениями каталога, постоянно считывая его содержимое. Но такой способ неэффективен с точки зрения потребления ресурсов. К счастью, в ядре Linux есть механизм, специально предназначенный именно для этой цели — для оповещения программы об изменениях в файловой системе. Он называется *inotify*.

Вот общая картина. Сначала вы создаете «экземпляр *inotify*» и получаете обратно дескриптор файла, точно так же, как если бы вы открыли файл методом `open()`. Затем вы добавляете «наблюдателя [watch]». По существу, этот «наблюдатель» говорит: «Я хотел бы отслеживать такой-то файл или такой-то каталог на предмет таких-то и таких-то изменений». Затем вы ждете оповещений об изменении, которые вы читаете из дескриптора файла (дальнейшее чтение заблокируется, пока сообщение не будет доставлено). Можно добавить сколько угодно наблюдателей и отслеживать отдельные файлы и каталоги, но при этом не работает рекурсия — т.е. при отслеживании каталога подкаталоги автоматически отслеживаться не будут.

Возьмемся за дело и обратимся к примеру. Вот небольшая программа, которая отслеживает два каталога (мой домашний каталог `home` и `/etc`) и выводит сообщение при каждом создании или удалении файла.

```
1 /* Отслеживаем изменения в директориях с помощью inotify */
2 #include <stdio.h>
3 #include <stdlib.h>
```

```
4 #include <sys/inotify.h>
5 #include <limits.h>
6 /* Буфер с объемом, достаточным для 100 событий */
7 #define BUFSIZE (100 * (sizeof(struct inotify_event) + NAME_MAX + 1))
8 void main()
9 {
10  int notifyfd, watchfd;
11  char eventbuf[BUFSIZE]; /* Сюда читаются события */
12  int n;
13  char *p;
14  struct inotify_event *event;
15  notifyfd = inotify_init(); /* Ошибки не проверяются */
16  watchfd = inotify_add_watch(notifyfd, "/home/chris", IN_CREATE | IN_DELETE);
17  watchfd = inotify_add_watch(notifyfd, "/etc", IN_CREATE | IN_DELETE);
18  /* Наши наблюдатели на месте, читаем поток событий */
19  while(1) {
20    n = read(notifyfd, eventbuf, BUFSIZE);
21    /* В цикле проходим события и сообщаем о них */
22    for (p = eventbuf; p < eventbuf + n; ) {
23      event = (struct inotify_event *) p;
24      p += sizeof(struct inotify_event) + event->len;
25      /* Выводим событие */
26      if (event->mask & IN_CREATE) printf("%s создано\n", event->name);
27      if (event->mask & IN_DELETE) printf("%s удалено\n", event->name);
28    }
29  }
30 }
```



События сообщений считываются из дескриптора `inotify`, содержащего имя файла переменной длины, поэтому для их перебора в буфере требуются хитрые конструкции с указателями.

Всё интересное начинается в строке 15, где мы инициализируем наш поток событий, получая обратно дескриптор файла. В строках 16 и 17 мы добавляем «наблюдателей» для нужных каталогов. Макросы `IN_CREATE` и `IN_DELETE` — это однобитовые значения, которые должны объединяться по «или» (как здесь и происходит) для указания событий, которые мы отслеживаем. Это лишь два из десятка возможных событий; полный список можно найти на `man`-странице `inotify`. После определения всех «наблюдателей» мы входим в цикл (строка 20), в котором считываем события об изменении из потока. Мы получаем одно или несколько сообщений о событиях, которые определяются в структуре `inotify_event`. В конце структуры есть поле переменной длины, которое содержит информацию о файле, вызвавшем событие (на рис. внизу слева). Это усложняет обработку событий в буфере, требуя явных операций с указателями (строка 24).

Для указания произошедшего события используются те же значения битов в поле `mask` структуры `inotify_event`, которые

мы изначально указали для выбора интересующих нас событий. В строках 26 и 27 проверяются два события, которые мы отслеживаем, и для каждого случая выводится соответствующее сообщение.

Легко представить, что наряду с использованием в графическом файловом менеджере *inotify* можно применить для получения следа аудита изменений в важных файлах.

Обработка аргументов команд

Сменим тему. Конечно же, вы знакомы с передачей аргументов программе в командной строке. Если вам когда-либо приходилось писать скрипты, то вы знаете, что эти аргументы доступны с помощью специальных переменных \$1, \$2 и т.д. Но как получить их в программе на C? Например, их можно получить с помощью двух аргументов, передаваемых функции `main()`. Разумеется, эти аргументы можно назвать как угодно, но вы станете нарушителем конвенции, если не назовете их `argc` и `argv`.

Первый аргумент, `argc`, просто сообщает нам количество аргументов. Вторым, `argv`, указывает на массив указателей, которые, в свою очередь, указывают на строки, содержащие аргументы. Обратите внимание, что в список включено имя программы в том виде, как она была вызвана (оно доступно как `argv[0]` и включено в количество аргументов `argc`), см. рис. внизу справа.

Поэтому можно перебрать аргументы командной строки в простом цикле:

```
int main(int argc, char *argv[])
{
    int i;
    for (i=0; i<argc; i++)
        printf("%s\n", argv[i]);
}
```

Часто нужна проверка, что пользователь указал должное количество аргументов. Вот пример программы, которая ожидает получить только одно имя файла:

```
int main(int argc, char *argv[])
{
    char *filename;
    if (argc != 2) {
        fprintf(stderr, "usage %s file\n", argv[0]);
        exit(1);
    }
    filename = argv[1];
    /* и так далее */
}
```

Обратите внимание на использование `argv[0]` для включения имени программы в сообщение "usage [использовано]". Время от времени вы будете встречать программы, которые анализируют `argv[0]` и ведут себя по-разному в зависимости от того, как они были вызваны. Раньше этим пользовались чаще, чем сейчас, но даже в своей системе я смог найти `unzip` и `zipinfo`, ссылающиеся на один и тот же исполняемый файл. А запустив команду

```
$ find /bin /usr/bin -type l -ls
```

вы увидите несколько символических ссылок на программы, которые (вероятно) используют тот же прием.

Обработка с `getopts()`

Вся эта история с `argv` выглядит не так уж плохо, если вы умеете работать с массивами указателей. Но все может усложниться. Взгляните на команду

```
ls -a -l foo bar
```

Аргументы, начинающиеся с "-", принято считать ключами, они же — опции команд. Конечно, их можно выделить, немного

поработав со строками. Не проблема! Потом мы понимаем, что придется разбираться и с эквивалентным синтаксисом:

```
ls -al foo bar
```

которым все пользуются, а также с

```
ls foo bar -a -l
```

которым пользуются мало, но он также правомерен. Обработка строк усложняется. И тут возникает путаница: опции идут последними, но обработать их нужно первыми. Также подумайте о команде

```
gcc -o demo demo.c
```

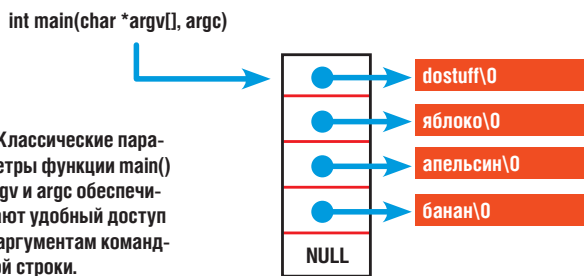
в которой опция **-o** объединяется с аргументом **demo** для указания выходного файла. Ну да, да, можно еще пообработать строки; но вариантов уже слишком много, и, сказать по правде, C не так уж искусно обращается со строками. Однако не всё так грустно. Нам поможет библиотечная функция под названием `getopts()`, которая возьмет на себя большую часть усилий по обработке аргументов командной строки. В подробностях это несколько запутанно, поэтому позвольте описать общую картину. Вы повторно вызываете функцию `getopts()` со списком аргументов `argv`, и она возвратит буквы опций по одной. При наличии опций, которые принимают следующие за ними аргументы, функция вернет и их тоже. Она также изменит порядок аргументов `argv`, так что в конце будут аргументы без опций, и сообщит, где в списке аргументов начинаются аргументы без опций:

```
1 /* Обработка аргументов командной строки с помощью
   getopt() */
2 /* Пример: argdemo -a -b -n 400 -t лиловый */
3 #include <unistd.h>
4 #include <stdio.h>
5 #include <string.h>
6 void main(int argc, char *argv[])
7 {
8     int c;
9     int aflag = 0;
10    int bflag = 0;
11    int numoption = 0;
12    char txtoption[200] = "default";
13    opterr = 0;
14    while ( (c = getopt(argc, argv, "abn:t:")) != EOF) {
15        switch (c) {
16            case 'a':
17                aflag = 1;
18                break;
19            case 'b':
20                bflag = 1;
21                break;
22            case 'n':
23                num option = atoi(optarg);
24        }
25    }
26 }
```

Скорая помощь

Переменная окружения `POSIXLY_CORRECT` используется для принудительного обеспечения совместимости с POSIX и преимущественно используется в ситуациях, «когда POSIX расходится со здравым смыслом». Например, если переменная `POSIXLY_CORRECT` установлена, по обнаружении первого аргумента без опций аргументы с опциями `getopts()` искать перестанет.

`$ dostuff яблоко апельсин банан` ← Командная строка



» Не хотите пропустить номер? Подпишитесь на www.linuxformat.ru/subscribe/

Скорая помощь



Если вам понадобится передать аргумент, который начинается с '-', но не должен рассматриваться как опция, воспользуйтесь аргументом '--', который отключает обработку опций для оставшейся части командной строки. Например, `$ argdemo -a -- -b` распознает `-a` как опцию, но считает `-b` аргументом без опций.

```

24 break;
25 case 't':
26 strcpy(txtoption, optarg);
27 break;
28 case '?':
29 fprintf(stderr, "неверная опция: -%c\n", optarg);
30 }
31 }
32 if (aflag) printf("Параметр a установлен\n");
33 if (bflag) printf("Параметр b установлен\n");
34 printf("Числовое значение опции %d\n", numoption);
35 printf("Текстовое значение опции %s\n", txtoption);
36 argv += optind; /* Подправляем для сдвига по опциям */
37 argc -= optind;
38 while (argc--) { /* Печать неопционных параметров */
39 printf("arg: %s\n", *argv++);
40 }
41 }

```

Комментарий в строке 2 отражает синтаксис команды, которую мы пытаемся обработать, `-a` и `-b` — простые булевы опции, `-n` — числовая опция, `-t` — текстовая опция. В строке 14 начинается цикл получения параметров. Мы передаем ему наши `argc` и `argv`, а третий аргумент ("`abn:t:`" в нашем примере) сообщает `getopt`, что `a`, `b`, `n` и `t` являются правильными опциями и что за `-n` и `-t` должны следовать их собственные аргументы. Возвращая буквы опций по мере их обнаружения, `getopt()` пробирается по командной строке и сдвигает все аргументы, не имеющие опций, в конец массива `argv`.

В блоках `case` выражения `switch` обрабатываются отдельные опции (строки 15–27). Параметры `-a` и `-b` просто устанавливают булевы флаги `aflag` и `bflag` (видимо, для дальнейшего использования в программе). Для параметров `-n` и `-t` `getopt()` помещает их аргументы в глобальную переменную `optarg`, которую мы извлекаем в виде числа (строка 23) или строки (строка 26). Последний блок `case` (строка 28) сработает, если `getopt()` встретит неожиданную опцию (например, "`x`").

Когда `getopt()` больше не может найти ни одной опции, она возвращает EOF, и мы выходим из цикла. В этот момент все аргументы без опций находятся в конце массива `argv`, и индекс первого из них записан в глобальную переменную `optind`. Поэтому мы изменяем значения `argc` и `argv` на этот индекс (строки 36, 37) и затем перебираем аргументы без опций (строки 38, 39).

Обработка «длинных» опций

Думаю, что это ребята из GNU придумали «длинные опции» в Linux. У некоторых из них есть традиционные однобуквенные эквиваленты (например, `ls --all` эквивалентно `ls -a`), а у некоторых — нет (например, `ls --color`). Предполагаю, что причиной появления длинных опций было то, что многим командам однобуквенных параметров не хватало (например, у компилятора C GNU опций более 400). К тому же в скриптах длинные опции более наглядны по сравнению со своими загадочными однобуквенными родичами.

Длинные опции можно обработать с помощью функции `getopt_long()`

библиотеки GNU C, но это немного скучно. Для каждой опции нужно заполнить структуру `option`, содержащую имя опции и некоторые флаги, определяющие наличие аргументов у опции и то, как они должны обрабатываться. Затем вы передаете функции массив таких структур. Модуль Python `parseargs` принимает длинные опции чуть более органично — они добавляются так же, как и любые другие, например:

```

p.add_argument("-a", action="store_true",
dest="aflag")
p.add_argument("--all", action="store_true",
dest="aflag")

```

Обратите внимание, что здесь используется подход, больше похожий на C (увеличение указателя в вызове `printf()`), а не на Fortran (увеличение целочисленного индекса массива, который я протестировал ранее).

Давайте рассмотрим эту программу в действии:

```

$ argdemo -a яблоко апельсин
Установлена опция a
Значение числовой опции 0
Значение текстовой опции принято по умолчанию
arg: яблоко
arg: апельсин
$ argdemo -n 500 -ab
Установлена опция a
Установлена опция b
Значение числовой опции 500
Значение текстовой опции принято по умолчанию
$ argdemo банан груша -t зеленый
Значение числовой опции 0
Значение текстовой опции зеленый
arg: банан
arg: груша

```

Разберите эти три примера сами. Обратите внимание, что `-ab` аналогично `-a -b`, а последний пример показывает, что параметры распознаются правильно, даже если идут после аргументов без параметров.

Обработка аргументов в Python

В Python есть список под названием `sys.argv`, который служит той же цели, что и `argv` в нашей программе на C. Вот версия программы, которую я показывал раньше, на Python. Ее аргумент — одно имя файла:

```

import sys
if (len(sys.argv) != 2):
    print("usage", sys.argv[0], "file")
    raise SystemExit(1)
filename = sys.argv[1]
# и т.д.

```

Обратите внимание, что прямого эквивалента `argc` в Python нет. Он и не нужен, так как длина таких списков, как `sys.argv`, в Python легко определяется.

В Python также есть модуль под названием `getopts`, который очень похож на функцию `getopts()` из стандартной библиотеки C, с которой мы только что познакомимся. Кроме него, есть модуль `argparse`, который предоставляет гораздо более высокоуровневую поддержку. Вот программа, которую я показал раньше (более или менее), переписанная с использованием модуля `argparse` из Python:

```

1 #!/usr/bin/python3
2 import argparse
3 p = argparse.ArgumentParser()
4 p.add_argument("-a", action="store_true", dest="aflag")
5 p.add_argument("-b", action="store_true", dest="bflag")
6 p.add_argument("-n", action="store", type=int,
dest="numoption")
7 p.add_argument("-t", action="store", dest="txtoption")
8 p.add_argument('files', nargs='*')
9 p.set_defaults(aflag=False, bflag=False, numoption=0,
txtoption="default")
10 args = p.parse_args()
11 if (args.aflag): print("Установлена опция a")
12 if (args.bflag): print("Установлена опция b")
13 print("Числовое значение опции ", args.numoption)

```

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

```
14 print("Значение текстовой опции ", args.txtoption)
15 print(args.files)
```

Давайте бегло пройдемся по коду. В строке 3 мы создаем ArgumentParser; этот парень и делает всю трудную работу. Строки 4–8 сообщают парсеру о правильных опциях команды. Например, строка 6 велит парсеру ожидать опцию `-n` и сохранить соответствующее целочисленное значение в `numoption`. Вся работа происходит в вызове `parse_args()` в строке 10. Мы получаем объект пространства имен (`args`), внутри которого хранятся переменные, содержащие значения обработанных аргументов командной строки, проверяем эти строки в строках 11–14 и затем выводим аргументы без опций в строке 15.

По структуре эта программа во многом напоминает версию на C, которую мы видели раньше. Преимущество ArgumentParser станет вам очевидным, если мы преднамеренно укажем неверную опцию:

```
$ ./argdemo.py -x
usage: argdemo.py [-h] [-a] [-b] [-n NUMOPTION] [-t TXTOPTION]
[files [files ...]]
argdemo.py: error: unrecognized arguments: -x
```

Произошло чудо. Без всякого программного содействия с моей стороны Python сформировал подробное сообщение 'usage' и сообщил о нераспознанной опции `-x`. Вы также бесплатно получили опции `-h` и `--help`, и их можно улучшить, добавив описание для каждого аргумента. Советую вам прочесть документацию Python для `argparse`: она поможет убедиться, что Python — более высокоуровневый язык, чем C!

Окружение

Наконец, в этом месяце я познакомлю вас с окружением. Окружение есть у каждого процесса, и по сути это просто набор строк вида

```
NAME=VALUE
```

Отдельные переменные называются переменными окружения, и их имена традиционно записываются в верхнем регистре. Возможно, вы уже знакомы с этой концепцией, если попользовались оболочкой. Список переменных оболочки можно вывести командой `env`, а добавить новую переменную — командой `export`, например:

```
$ FOO=BAR
$ export FOO
```

Или сделать все одной командой:

```
$ export FOO=BAR
```

Уточним, что просто присвоить переменной оболочки значение *недостаточно* для того, чтобы она попала в окружение.

Обращаться к окружению изнутри наших программ можно двумя способами. Во-первых, с помощью глобальной переменной `environ`, указывающей на массив указателей, которые, в свою очередь, указывают на отдельные строки, почти так же как `argv` (см. внизу справа). Поэтому можно перебрать все переменные окружения следующим образом:

```
extern char **environ;
main()
{
    char **p;
    for (p=environ; *p != NULL; p++)
        printf("%s\n", *p);
}
```

Обычно нам нужно получить значение всего одной переменной, а это гораздо проще делается с помощью `getenv()`: например, `getenv("DISPLAY")` вернет указатель на значение переменной окружения `DISPLAY` или `NULL`, если такой переменной нет.

Среди других системных вызовов для работы с окружением имеются `setenv()` и `unsetenv()`, представляющие собой

Окружение и безопасность

Учтите, что программа никак не может управлять окружением, которое получает от родителя, и должна считать его «недоверенным вводом». Например, вот это выглядит довольно безопасно:

```
char homedirname[MAX_PATH];
strcpy(homedirname, getenv("HOME"));
```

В конце концов, `HOME` содержит путь к каталогу `home` и не может быть длиннее `MAX_PATH`, верно? Неверно! Вредоносный родительский процесс может присвоить `HOME` все, что угодно, поэтому вы подвержены атаке переполнения буфера. Программы, запускаемые с установленным битом `setuid`, должны

быть особенно осторожны, и есть процедура `secure_getenv()` (расширение GNU), которая отказывается получать переменные окружения в ситуациях, когда необходимо «защищенное выполнение». Подробности — на map-странице.

В частном случае с `HOME` вместо того, чтобы доверять значению, переданному оболочкой, безопаснее определить идентификатор текущего пользователя с помощью вызова `getuid()`, а затем вызвать `getpwuid()` с этим идентификатором для получения записи пользователя в файле паролей, которая (среди прочего) содержит домашний каталог пользователя.

эквиваленты команд `Bash export` и `unset`, и `clearenv()`, которая удаляет все окружение.

Смысл окружения состоит в том, что обычно оно передается дочерним процессам. Это создает удобный механизм для передачи конфигурационных параметров, которые должны быть доступны только тем программам, которые вы запускаете. Учтите, что это наследование работает только в одну сторону: изменения, вносимые в окружение дочерним процессом, не передаются родителю, а изменения, произведенные в окружении родителя после создания дочернего процесса, не передаются дочернему процессу.

Переменные окружения обычно используются для передачи программе небольшого количества конфигурационной информации. Многие программы опрашивают значение определенных переменных, и то же самое делают некоторые процедуры стандартной библиотеки C, например, программа `crontab`, которая позволяет изменить файл `crontab` и затем проверяет его синтаксис, используя переменную окружения `EDITOR`, чтобы определить, какой текстовый редактор вы предпочитаете. Поэтому команды:

```
$ export EDITOR=nano
$ crontab -e
```

откроют редактор `nano`. Также можно выполнить следующую команду:

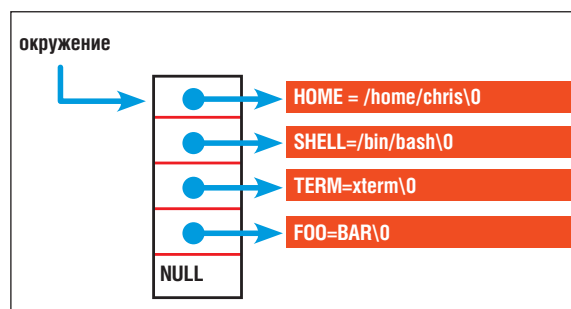
```
$ EDITOR=nano crontab -e
```

которая запустит `crontab` с переменной окружения `EDITOR`, но без изменения собственного окружения оболочки.

Другой пример — переменная окружения `TZ` (timezone, часовой пояс), используемая различными процедурами стандартной библиотеки, которая преобразует текущее время в локальное. Но подробности подождут до нашей встречи в следующем месяце, когда я поведу разговор о времени. Время — это категория философская! **LXF**

Скорая помощь

Получить окружение любого процесса, если вы знаете его идентификатор, можно, заглянув в файл `/proc/PID/environ`, где `PID` — идентификатор процесса. Записи заканчиваются символами нуля, а не переводами строки, поэтому читать их тяжко.



➤ По аналогии с `argv` окружение представляет собой массив строк, завершающихся символами `null`.

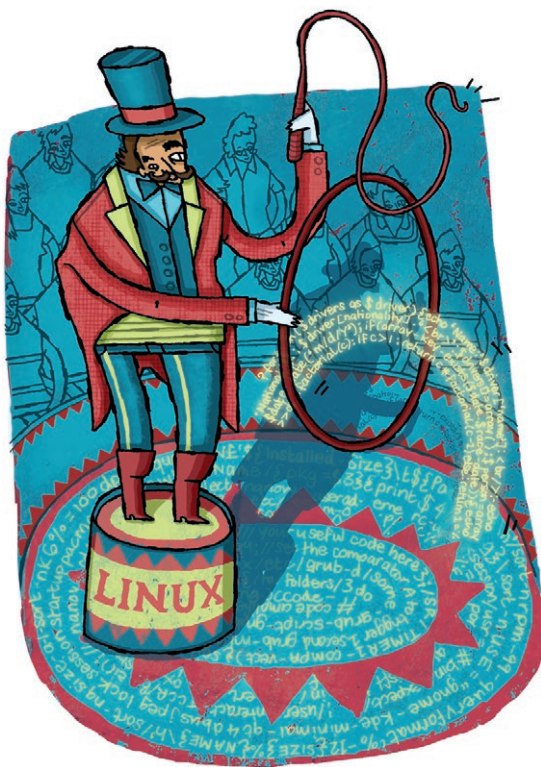
Scala: Просто и функционально

Михалис Цукалос объясняет базовые концепции, необходимые для понимания, как начать программировать на языке Scala.



Наш эксперт

Михалис Цукалос — администратор UNIX, программист, администратор баз данных и математик, а еще он любит писать статьи и изучать что-то новое.



```

2. mtsouk@mail: ~/docs/article/working/Scala.LXF/code (ssh)
mtsouk@mail:~/docs/article/working/Scala.LXF/code$ cat hw.sh
#!/bin/sh
exec scala "$@" "$@"
}#
object HW {
  def main(args: Array[String]) {
    println("Hello World!")
  }
}
HW.main(args)
mtsouk@mail:~/docs/article/working/Scala.LXF/code$ chmod 755 hw.sh
mtsouk@mail:~/docs/article/working/Scala.LXF/code$ ./hw.sh
Hello World!
mtsouk@mail:~/docs/article/working/Scala.LXF/code$ scala
Welcome to Scala version 2.9.2 (OpenJDK 64-Bit Server VM, Java 1.6.0_34).
Type in expressions to have them evaluated.
Type :help for more information.

scala> object HW {
  |   def main(args: Array[String]) {
  |     println("Hello World!")
  |   }
  | }
defined module HW

scala> HW.main(null)
Hello World!

scala> :q
mtsouk@mail:~/docs/article/working/Scala.LXF/code$

```

➤ Рис. 1. Код можно запустить в интерактивной среде Scala или включить в скрипт оболочки!

с компиляцией. Обычно в интерактивной среде можно попробовать что-то новое и поэкспериментировать, а с помощью компилятора сгенерировать исполняемые файлы. Для выхода из интерактивной среды Scala нужно набрать :q и нажать Enter.

Код Scala для программы “Hello World!” таков:

```

object HW {
  def main(args: Array[String]) {
    println("Hello World!")
  }
}

```

Сначала надо определить новый объект под именем HW — или любым именем на ваш вкус — в котором реализуется функция **main()**. В функции **main()** находится код, который выводит на экран желаемое сообщение.

Код на Scala можно запустить тремя способами: как скрипт, в интерактивной среде Scala и через компилятор. На рис. 1 показаны первый и второй способы. Третий способ работает так:

```

$ scalac hw.scala
$ ls -l
total 12
-rw-r--r-- 1 mtsouk mtsouk 586 Mar 16 09:41 HW.class
-rw-r--r-- 1 mtsouk mtsouk 581 Mar 16 09:41 HW$.class
-rw-r--r-- 1 mtsouk mtsouk 87 Mar 14 21:04 hw.scala
$ scala HW
Hello World!

```

Также ее можно запустить из командной строки UNIX в интерактивной среде, командой **scala fibo.scala**. Преимущество

На этом уроке мы расскажем о Scala, современном функционально-объектном языке программирования, разработанным Мартином Одерски. Название “Scala” происходит от слов “scalable [масштабируемый]” и “language [язык]” и означает, что язык может «расти» с потребностями пользователей. Мы представим удобные технологии и расскажем о характеристиках Scala, но не сможем описать все аспекты языка — их можно узнать только на практике, поэтому советуем как можно скорее начинать писать небольшие программы на Scala.

Примечание: если вы следите за нашими сериями статей о языках программирования, то, возможно, обнаружили, что в качестве примеров кода обычно используется определение чисел Фибоначчи, реализация алгоритма сортировки и ввод/вывод в файлы. Это сделано специально, чтобы вы сравнили, как похожие программы реализуются на разных языках.

Установка Scala

Установить Scala в Debian можно, выполнив команду **apt-get install scala** с правами root, а определить версию языка — выполнив команду **scala -version** после установки. В Scala есть утилита для выполнения кода Scala (**scala**), которая представляет собой интерактивную среду, и компилятор Scala (**scalac**), который поможет

Скорая помощь

Подробную информацию о Scala можно получить на www.scala-lang.org. Две хорошие книги — “Programming In Scala [Программирование на Scala]” Мартина Одерски [Martin Odersky] и “Scala In Action [Scala в действии]” Ниланджана Райчаудхури [Nilanjana Raychaudhuri].

данного метода над компилятором в том, что не создается никаких промежуточных файлов, но при этом он медленнее.

Расчет чисел Фибоначчи

Первая реализация программы для расчета чисел Фибоначчи такова:

```
def myFibo( n : Int) : Int = n match {
  case 0 | 1 => n
  case _ => myFibo( n-1 ) + myFibo( n-2 )
}
```

Строка **case _** применяется ко всем случаям, которые не покрываются первой строкой **case** выражения **match**. Важно охватить все случаи, иначе программа может завершиться с ошибкой. В файле **fibonacci.scala** можно найти полностью рабочий пример, в котором используется **myFibo**. Цикл **for**, используемый в **fibonacci.scala**, может иметь массу вариантов; в данном случае используется вариант с диапазоном. Другие формы включают использование коллекций и фильтров и ключевого слова **yield**.

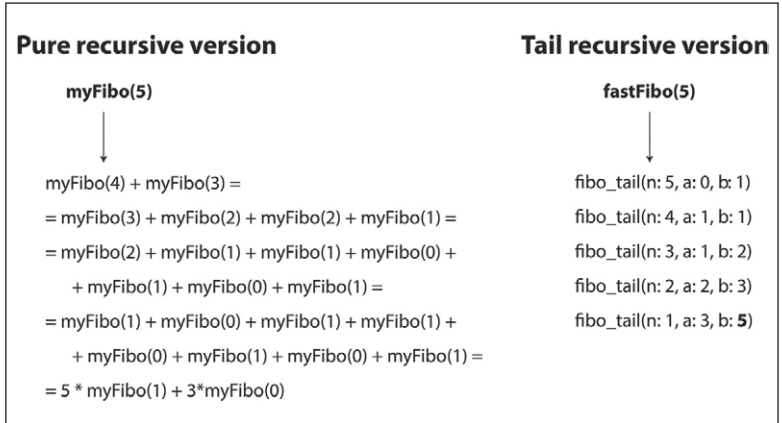
Проблема с предыдущей рекурсивной реализацией в том, что она очень медленно вычисляет большие числа Фибоначчи и может вызывать исключения переполнения стека, так как каждый раз, когда функция вызывает другую функцию — даже сама себя — необходимо сохранять состояние обеих функций до того момента, пока они не завершат свое выполнение. Попробуйте вычислить 50-е число Фибоначчи, и вы поймете, что я имею в виду. На рис. 2 показан объем вычислений, которые нужно выполнить для определения пятого числа Фибоначчи методом чистой рекурсии. При каждом вычислении необходимо много раз вызывать функцию **myFibo()**.

Теперь пора научиться вычислять числа Фибоначчи быстрее. Код функции **fastFibo()** на Scala таков:

```
def fastFibo( n : Int) : Int = {
  def fibo_tail(n: Int, a: Int, b: Int): Int = n match {
    case 0 => a
    case _ => fibo_tail(n-1, b, a+b)
  }
  return fibo_tail(n, 0, 1)
}
```

В этом методе используется хвостовая рекурсия. Она удобна тем, что хвостовые вызовы выполняются без добавления нового стекового фрейма в существующий стек вызовов. Вычисление 40-го числа Фибоначчи происходит почти мгновенно. Функция **fibo_tail** принимает три целочисленных параметра и возвращает целое число.

На рис. 2 также показано количество вычислений, необходимых для определения пятого числа Фибоначчи методом



► Рис. 2. Вычисление пятого числа Фибоначчи с помощью чистой рекурсии и хвостовой рекурсии.

хвостовой рекурсии. Выполняется всего один вызов функции **fastFibo()**. Различие между вызовами функций все равно велико, несмотря на тот факт, что для определения пятого числа Фибоначчи требуется не так уж много операций.

Реализация пузырьковой сортировки

Покажем, как реализовать пузырьковую сортировку в Scala традиционным методом итерации, поскольку Scala также поддерживает итеративное программирование. Это большой плюс: вы можете постепенно учиться функциональному программированию и в то же время решать задачи так, как вы уже умеете.

Полный код функции, реализующей пузырьковую сортировку, таков:

```
def bubbleSort(input: Array[Int]): Array[Int] = {
  val limit = input.size - 1
  for (a <- 1 to limit) {
    for (b <- limit to a by -1) {
      if (input(b) < input(b - 1)) {
        val x = input(b)
        input(b) = input(b - 1)
        input(b - 1) = x
      }
    }
  }
  input
}
```

В качестве входных параметров функция **bubbleSort()** принимает массив целых чисел и выводит также массив целых чисел. ►

Родство Scala и Java

Scala и Java связаны между собой, так как Scala работает на виртуальной машине Java Virtual Machine. Классы Scala — это по сути классы JVM, и вы можете из Scala прозрачно использовать объекты Java, вызывать их методы и наследовать классы Java. Как вы, наверное, можете представить, библиотеки Java можно использовать напрямую в коде Scala. Также в коде Java легко использовать классы и объекты Scala. На рис. 4 показана подробная информация, а также необходимый код. Даже если вы не большой поклонник Java, возможность вызывать код Scala

из программ на Java — огромное преимущество; ну, а не любите Java — не пользуйтесь этим!

При этом Scala поддерживает массу возможностей, не поддерживаемых Java, включая каррирование [преобразование функции с несколькими аргументами в несколько функций с одним аргументом], извлечение типов, неизменяемость, «ленивые» вычисления и сопоставление с шаблоном. Scala также обладает улучшенной системой типов, поддерживающей алгебраические типы данных, ковариантность и контрвариантность,

типы более высокого порядка и анонимные типы. Другие возможности Scala, не представленные в Java, включают перегрузку операторов, дополнительные параметры, именованные параметры, «сырые» строки и не проверяемые исключения.

Scala предназначен для улучшения программирования на Java и не прикован к недостаткам Java. Единственное, чего в Scala не исправить — тот факт, что на выполнение программы, использующей JVM, нужно больше времени, чем на выполнение двоичного файла в Linux.

► Не хотите пропустить номер? Подпишитесь на www.linuxformat.ru/subscribe/!

Если вам знаком алгоритм пузырьковой сортировки, то нет проблем с пониманием его реализации в Scala. Поскольку каждую новую переменную в Scala нужно определять с ключевым словом **var**, следует определить переменную для нового массива с помощью ключевого слова **var** посредством кода **var myArray = createArray(10)**. Если определить переменную с ключевым словом **val** вместо **var**, вы не сможете впоследствии изменить ее значение (это будет константа). Функция **scala.util.Random** используется внутри **createArray()** для заполнения массива случайными числами перед их сортировкой.

Полностью рабочую программу с пузырьковой сортировкой вы найдете в файле **bubble.scala**.

Аргументы командной строки

Теперь поучимся обрабатывать аргументы командной строки в Scala. Из определения функции **main()** вы уже знаете, как с помощью массива **args** передаются аргументы командной строки. Теперь вы научитесь получать их и оперировать ими в своих программах.

Простейший способ вывести на экран переданные аргументы командной строки — воспользоваться строкой кода **args foreach println** в своей программе. Эта команда использует выражение **foreach** для получения всех элементов массива **args** и выводит их на экран командой **println**.

Следующий код проверяет все аргументы командной строки и выводит те из них, которые являются положительными целыми числами:

```
for ( x <- args ) {
  if (isAllDigits(x)) {
    println(x + " положительное число!")
  }
}
def isAllDigits(x: String) = x.map(Character.isDigit(_)).
reduce(_&&_)
```

Всю черную работу делает функция **isAllDigits()**, которая с помощью **map** и **reduce** проверяет, состоит ли переданная в качестве аргумента строка исключительно из цифр. Чтобы увидеть

код полностью, обратитесь к файлу **args.scala**. Так как **args** — это обычный массив, правильность количества аргументов можно проверить следующим образом:

```
if ( args.size != 2 ) {
  println("Аргументов командной строки д.б. ровно два.")
  System.exit(1)
}
```

Вызов **System.exit(1)** используется для немедленного завершения программы.

Возможность получить данные от пользователя очень удобна. Посмотрим, как это делается в Scala. Следующий код просит пользователя ввести данные и сохраняет их в переменной под именем **userName**:

```
val userName = readLine("Пожалуйста, назовите свое имя: ")
```

Функция **readInt()** позволяет прочесть целое число, а чтобы известить пользователя о необходимости что-то ввести, служит **println()**. Получив не целое число, **readInt()** выдаст исключение.

Ввод/вывод в файлы в Scala

Пора написать простую программу, которая прочтет текстовый файл строка за строкой, вставит в начало каждой строки номер и сохранит результат в новом текстовом файле. Оба файла указываются в аргументах командной строки программы.

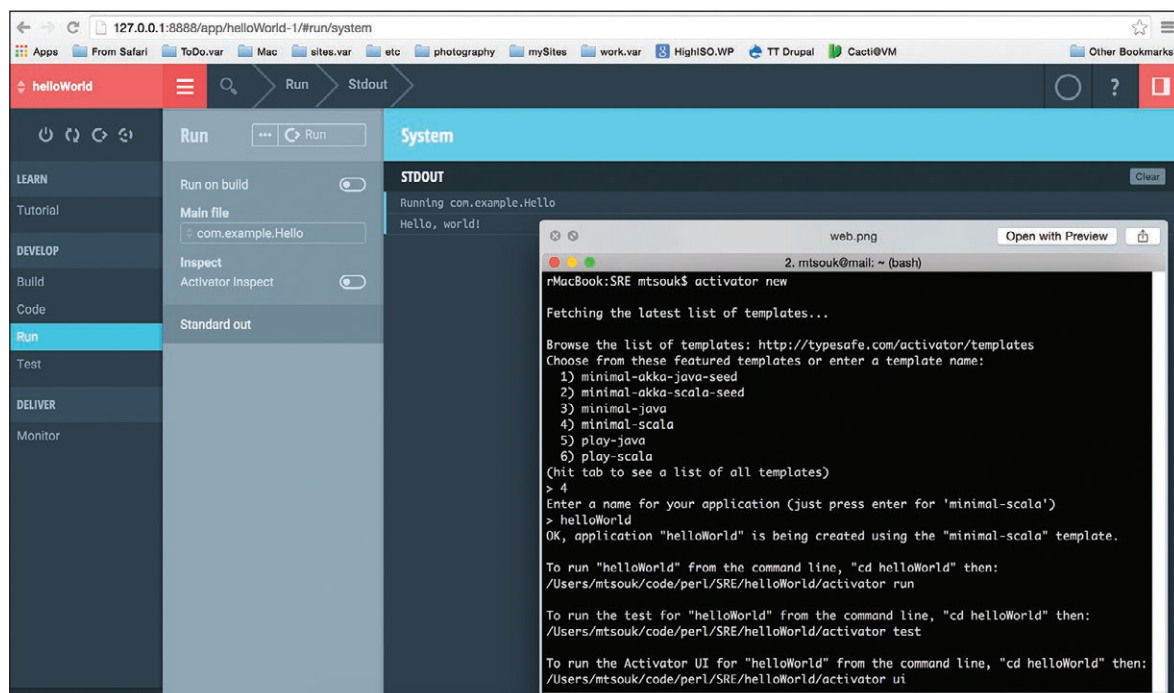
Пройти по содержимому текстового файла строка за строкой можно следующим образом:

```
import scala.io.Source
for (line <- Source.fromFile(input).getLines()) {
  println(line)
}
```

Строка **import scala.io.Source** велит Scala импортировать класс **scala.io.Source**, который необходим для чтения текстового файла. Здесь используется цикл **for**, но при желании вместо него можно использовать **foreach**.

Для записи в текстовый файл воспользуйтесь следующей функцией:

```
def appendFile(fileName: String, line: String) = {
  val fileToWrite = new FileWriter(fileName, true)
```



➤ Рис. 3. Вывод простого web-приложения, разработанного в Scala с использованием фреймворка Play.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Сопоставление с шаблоном в Java

Как и в любом современном языке программирования, в Scala играет ключевую роль сопоставление с шаблоном. Оно может сделать ваш код более элегантным, умным и удобным для восприятия, так как позволяет писать код с помощью альтернативных конструкций в соответствии с потребностями и с задачами, которые пытается решить разработчик. Ниже показан код для сопоставления всех целых чисел:

```
def testForInteger(x: Int): String = x match {
  case 1 => "один"
  case 3 => "три"
```

```
  case _ => "ПРОЧЕЕ"
}
```

Сопоставление с шаблоном не обязано быть простейшим, в чем можно убедиться на таком примере кода:

```
def whatIsIt(x: Any): String = x match {
  case i: Int => "Целое число: " + i
  case d: Double => "Число с двойной точностью: " + d
  case s: String => "Это строка: " + s
  case _ => "Неизвестно!"
}
```

Порядок условий сопоставления имеет значение, поскольку выполняется код только для первого соответствия. Если в ваш код на Scala вкралась логическая ошибка, вы можете получить предупреждение, говорящее: "Unreachable code due to variable pattern [Этот код никогда не выполнится из-за шаблона сопоставления переменной]".

Отличным упражнением для вас станет попытка написания простой версии *grep* на Scala! Не стесняйтесь сразу же реализовать все параметры *grep*, просто следите, чтобы программа была стабильной и в ней не было ошибок.

```
try {
  fileToWrite.write(line + "\n")
}
finally fileToWrite.close()
}
```

Для записи в текстовый файл придется использовать Java, так как в Scala нет хорошо отработанной библиотеки ввода/вывода. Поскольку функция `fileToWrite.write()` автоматически не добавляет символ перевода строки в конце каждой строки, нужно добавить его вручную. Если выходного файла не существует, он создается автоматически. А если он существует, все новые строки будут добавлены в конец файла.

Чтобы увидеть код Scala во всем его великолепии, обратитесь к файлу `file.scala`. Из этого файла вы также узнаете, как с помощью кода на Java проверить, существует ли файл, перед тем, как попробовать его прочесть!

Функциональное программирование

Возможно, вы заметили, что самый свежий тренд в языках программирования — функциональное программирование. Scala — не исключение из этого неформального правила, и мы раскроем эту тему подробнее, представив два практических примера. Первый касается использования анонимных (безымянных) функций, второй — использования функций как переменных.

В следующем коде определяется список и затем выводятся только четные числа из этого списка, с помощью анонимной функции:

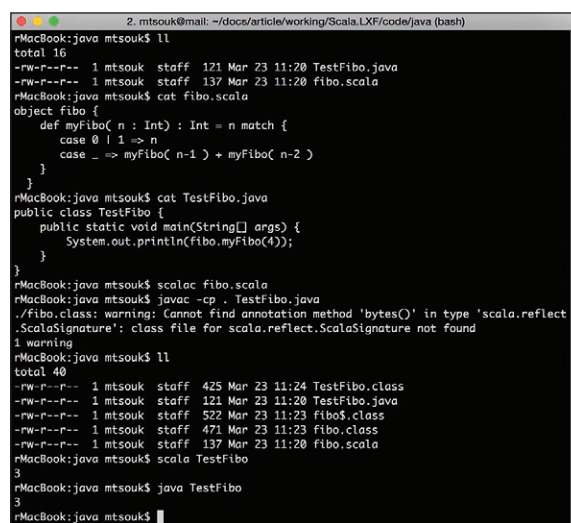
```
scala> val myList = List(1, 3, 4, 5, 6, 0, 1)
myList: List[Int] = List(1, 3, 4, 5, 6, 0, 1)
scala> val odds = myList.filter((i: Int) => i % 2 == 1)
odds: List[Int] = List(1, 3, 5, 1)
```

Код безымянной функции, которая используется для фильтрации ввода, таков: `(i: Int) => i % 2 == 1`. Обозначение `=>` можно считать своего рода преобразователем, так как оно преобразует то, что находится слева от него, с использованием алгоритма, указанного справа от него. Преимущество в том, что не нужно писать для этого отдельной функции! Так можно профилировать любой массив.

Также можно назначить анонимную функцию переменной: конструкция `var mySquare = ((i: Int) => i * i)` определяет переменную с именем `mySquare`, которая возвращает квадрат целого числа. После этого `mySquare()` можно пользоваться как обычной функцией. Увидеть оба случая в действии можно, заглянув в файл `funprog.scala`.

Web-приложения

Сегодня почти в каждом современном языке программирования есть средства для разработки web-приложений, и Scala здесь



```
2 mtsouk@mail: ~/docs/article/working/Scala.LXF/code/java (bash)
rMacBook:java mtsouk$ ll
total 16
-rw-r--r--  1 mtsouk  staff  121 Mar 23 11:20 TestFibo.java
-rw-r--r--  1 mtsouk  staff  137 Mar 23 11:20 fibo.scala
rMacBook:java mtsouk$ cat fibo.scala
object fibo {
  def myFibo(n: Int): Int = n match {
    case 0 | 1 => n
    case _ => myFibo(n-1) + myFibo(n-2)
  }
}
rMacBook:java mtsouk$ cat TestFibo.java
public class TestFibo {
  public static void main(String[] args) {
    System.out.println(fibo.myFibo(4));
  }
}
rMacBook:java mtsouk$ scalac fibo.scala
rMacBook:java mtsouk$ javac -cp . TestFibo.java
./fibo.class: warning: cannot find annotation method 'bytes()' in type 'scala.reflect.ScalaSignature': class file for scala.reflect.ScalaSignature not found
1 warning
rMacBook:java mtsouk$ ll
total 40
-rw-r--r--  1 mtsouk  staff  425 Mar 23 11:24 TestFibo.class
-rw-r--r--  1 mtsouk  staff  121 Mar 23 11:20 TestFibo.java
-rw-r--r--  1 mtsouk  staff  522 Mar 23 11:23 fibo$.class
-rw-r--r--  1 mtsouk  staff  471 Mar 23 11:23 fibo.class
-rw-r--r--  1 mtsouk  staff  137 Mar 23 11:20 fibo.scala
rMacBook:java mtsouk$ scala TestFibo
3
rMacBook:java mtsouk$ java TestFibo
3
rMacBook:java mtsouk$
```

► Рис. 4. Использование, компиляция и запуск кода Scala из программы на Java.

не является исключением! Покажем, как с помощью Scala написать маленькое приложение. Два самых известных web-фреймворка Scala — это Lift (<http://liftweb.net>) и Play (<https://www.playframework.com>), который основан на Java NIO и хорошо масштабируется. Основное различие между ними в том, что Lift запоминает состояние, а Play — нет, по аналогии с Rails и Django.

На сайте Play есть информация об установке фреймворка на ваш компьютер. После установки нужно создать новый проект командой `activator new`. Сделав это, выберите тип проекта `minimal-scala`. Назовите проект `helloWorld`, при этом создастся новый каталог с таким же именем.

Если набрать `.activator ui` в командной строке из каталога проекта, будет запущено много процессов и выведено много данных. Самая интересная часть вывода — вот эта:

```
Play server process ID is 20053
[info] play - Application started (Prod)
[info] play - Listening for HTTP on /127.0.0.1:8888
[info] a.e.s.Slf4jLogger - Slf4jLogger started
[info] application - onStop received closing down the app
```

Файл с кодом Scala — `src/main/scala/com/example/Hello.scala`. Чтобы увидеть приложение по умолчанию, наберите в адресной строке `http://localhost:8888`.

На рис. 3 показан вывод web-приложения, а также все рабочее окружение. Однако более подробный экскурс в разработку web-приложений для фреймворка Play выходит за рамки нашей статьи.

Замечательный способ изучить Scala — реализовать на нем традиционные утилиты UNIX. Можете начать с простых утилит и перейти к более сложным, таким как `ls`, `rm`, `df` и `mv`, по мере накопления опыта. **LXF**



Unity 5: Пишем 3D-игру в Linux

Робин де Йонг предлагает вам пошаговое руководство по созданию своей первой игры с помощью популярного и бесплатного SDK для создания игр.



Наш эксперт

Робин де Йонг написал и выступил редактором нескольких книг о Unity 3D, включая предстоящую "Unity in Action", а также является автором нескольких программ о трехмерном ПО и играх.

Платформа Linux много лет была обделена играми, но эти мрачные времена прошли. Правда, разработка игр для Linux немного отстает от общих трендов; даже Unity, популярная среда разработчика, из которой экспортировано в Linux сотни игр, все еще не имеет «родной» поддержки.

Поэтому мы обращаемся к Unity скрепя сердце; она поддерживает Linux как движок, но не как платформу для разработки. Конечно, мы хотим видеть в Linux больше игр, и вот приходится использовать SDK на Windows. К счастью, Unity 5 запускается в Linux через Wine. На Mageia 4 мы с этим изрядно помучились, но с Ubuntu или Linux Mint проблем не было. Однако учтите, что пользователи Ubuntu и его производных столкнутся с фатальной ошибкой, если определенный флаг не установлен в ноль. Для этого выполните следующую команду с правами root:

```
$echo 0 | sudo tee /proc/sys/kernel/yama/ptrace_scope
```

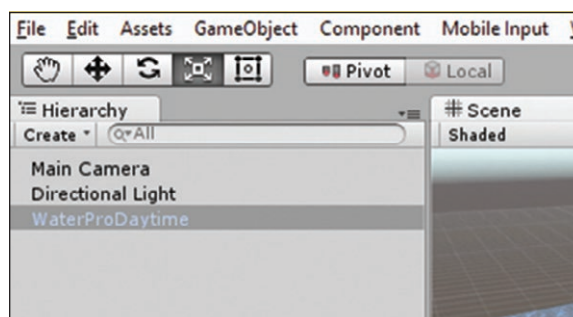
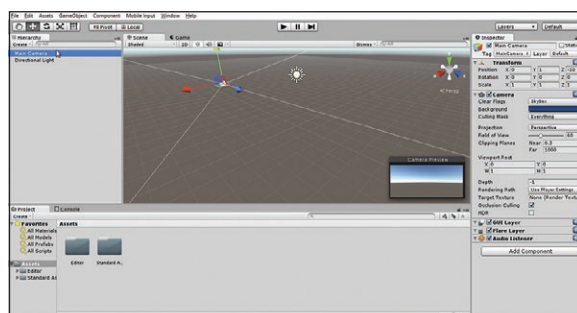
Чтобы установить и настроить Unity 5, установите через центр управления программами (Ubuntu) PlayOnLinux. При первом запуске PlayOnLinux может задать несколько вопросов — например, о доустановке шрифтов. Просто примите все изменения, и установка продолжится.

Установщик Unity для Windows загружается с <http://unity3d.com/unity/download>. Unity устанавливается скриптом PlayOnLinux, который можно взять с GitHub: <http://bit.ly/LXFunity>.

В PlayOnLinux выберите Tools > Run a local script [Инструменты > Запустить локальный скрипт] и выберите свежезагруженный файл Unity.pol. Следуйте указаниям установочного скрипта. Так как скрипт неофициальный, сначала вы должны принять на себя ответственность за его запуск. Скрипт устанавливает Wine и подборку компонентов Windows, которые занимают на диске несколько гигабайт. Появится странноватое диалоговое окно, которое потребует принять (или восстановить) установку (подозреваем, что в Mageia эти окна отображаются неправильно, вызывая ошибки). Можете попить чайку, после чего вы вернетесь к выбору установщика Windows и перейдете к установке Unity 5.

Не обойтись без еще одной чашечки чая; и после успешной установки создайте свою учетную запись Unity на сайте Unity.

Если ваши окна расположены иначе, выберите Window > Layouts > Default [Окна > Расположение > По умолчанию].



В Unity есть базовая панель преобразований — слева направо: View [Просмотр], Translate [Преобразование], Rotate [Поворот], Scale [Масштабирование], Select [Выбор].

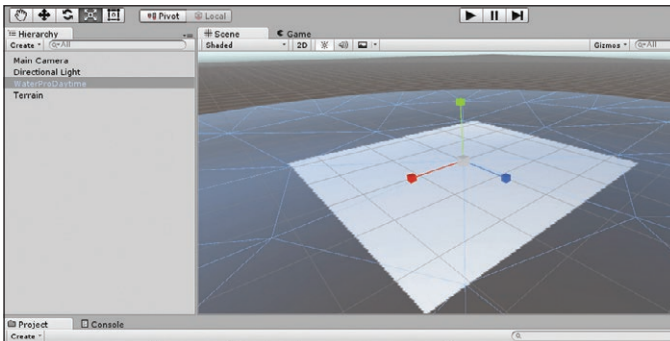
Теперь вы можете запустить Unity с рабочего стола или PlayOnLinux, выбрать и создать бесплатную учетную запись Unity, зайти в нее, и все будет готово.

Откройте Unity и выберите New Project [Новый проект]. Назовите свой проект "IslandRun" и в том же окне нажмите кнопку Asset Packages [Пакеты ресурсов]. Поставьте галочки Cameras [Камеры], Characters [Символы], Environment [Окружение] и Vehicles [Автомобили], нажмите Done [Готово] и затем Create Project [Создать проект]. Библиотека ресурсов игры некоторое время будет загружаться, и ход ее загрузки будет отображаться на прогресс-индикаторе. Оказавшись в редакторе Unity Editor, вы увидите окно, подобное любому пакету 3D-моделирования. Вверху находится главная сцена, где помещаются трехмерные или двумерные объекты, создаваемые с помощью программ вроде GIMP или Blender и затем импортируемые, но на нашем уроке мы воспользуемся готовыми ресурсами из библиотек, которые вы только что загрузили.

Устраиваем сцену

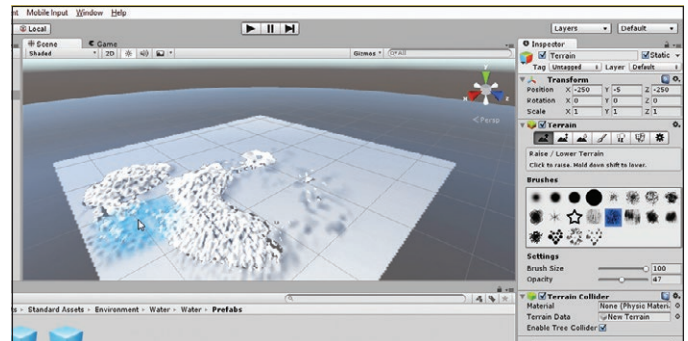
Окно Scene [Сцена] можно представить себе как киносцену, где вы предвзительно собираете компоненты своей игры. По щелчку на вкладке Game [Игра] над сценой в окне появится игровое представление. Продолжая аналогию с киносценой, в игровом представлении вы видите только то, что увидит аудитория через камеры, расположенные на сцене. Прежде чем продолжить, нажмите на вкладку Scene. Рядом с этими окнами слева есть список всего находящегося в данный момент на сцене. Представьте, что это список рабочего сцены. При щелчке по камере в списке она будет выделена на сцене. Если набрать cam в поле поиска, то в списке останется только то, что содержит эти буквы — так очень удобно находить нужные компоненты сцены, например, камеры. Справа от окна расположено окно Inspector [Инспектор], где будет показана подробная информация о выбранной камере. »

Снабжаем остров поверхностью



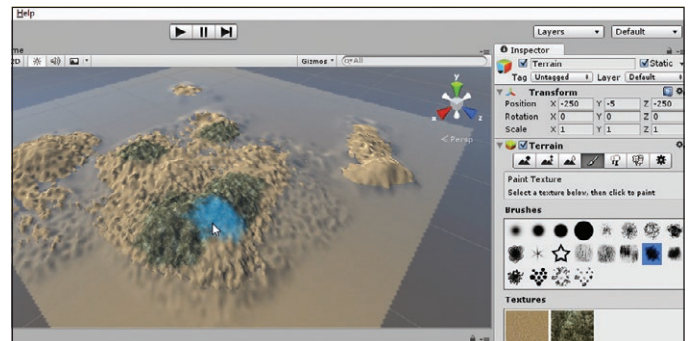
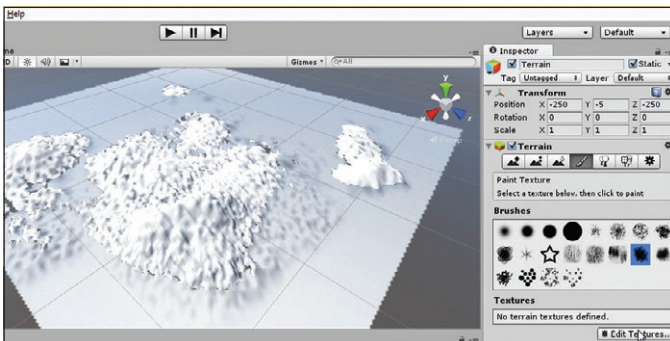
1 Создаем поверхность посреди воды

В Unity есть инструменты редактирования поверхностей, и мы воспользуемся ими для создания острова. Выберите Game Object > 3d Object > Terrain [Объект игры > Трехмерный объект > Поверхность]. В окне Inspector введите X -250, Y -5 и Z -250. Тогда остров будет расположен в центре воды, которую мы уже создали в основной статье [см. стр. 87], так как по умолчанию поверхность имеет форму квадрата с шириной 500. -5 по оси Y погружают поверхность ниже уровня воды.



2 Рисуем остров

На этом этапе подольем воды, чтобы она простиралась за пределы нашей поверхности. Выберите ее и введите X 1000 и Z 1000. Теперь приподнимем различные части земли, чтобы получился остров. Выбрав поверхность, нажмите кнопку Raise/Lower Terrain [Поднять/опустить поверхность]. Выберите одну из крупных кистей и установите ее размер в 100, а Opacity [Прозрачность] в 50. Нарисуйте на поверхности общие очертания нашего острова.



3 Добавляем детали

Нарисуйте на поверхности общую форму нашего острова. Уменьшите размер и прозрачность кисти и повторите форму со средней детализацией. Наконец, воспользуйтесь функцией Smooth Height [Сглаживание высоты], если остров слишком неровный. Кнопка прокрутки меняет масштаб, а средняя кнопка мыши прокручивает вид. Нажмите кнопку Paint Texture [Нарисовать текстуру] и выберите Edit Textures [Изменить текстуры] и Add Texture [Добавить текстуру].

4 Рисуем и применяем текстуры

Нажмите Select [Выбрать] на квадрате слева под названием Albedo [RGB]. Прокрутите вниз, найдите SandAlbedo [Текстура песка] и дважды щелкните по нему. Нажмите Add [Добавить]. Обратите внимание, что песок появится на всей поверхности. Теперь точно так же добавьте еще одну текстуру, присмотрев любую из текстур Cliff [Скала] или Rocky [Камень]. Затем выберите ее и воспользуйтесь кистью так же, как прежде, нарисовав камни там, где они должны проглядывать сквозь песок.

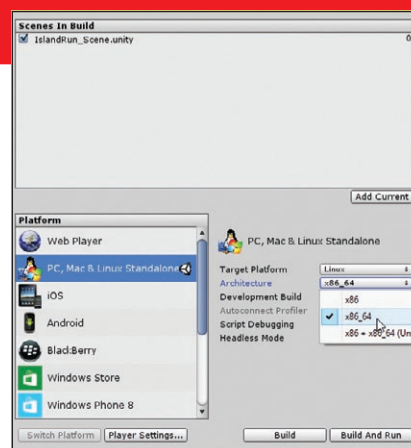
Экспорт игры в Linux

Теперь ваша игра должна быть готова к тому, чтобы упаковать ее в пакет и попробовать на своем компьютере. Сохраните сцену и затем сохраните проект. Теперь выберите File > Build Settings [Файл > Настройки сборки]. Выберите PC Mac & Linux Standalone [Настольные компьютеры Mac и Linux]. Выберите Linux в качестве Target Platform [Целевой платформы] и архитектуру x86 для 32-битных компьютеров или x86_64 для 64-битных компьютеров. Выберите Add Current [Добавить текущее], чтобы добавить к сборке свою сцену с островом. Нажмите Player Settings [Настройки проигрывателя]. Введите название компании и другую необходимую информацию,

такую как экран заставки. Нажмите Build [Собрать], введите имя файла и нажмите Save [Сохранить].

При копировании файла в Linux помните, что напрямую с флешки запускать его не стоит — это вряд ли получится. Скопируйте файл на рабочий стол и убедитесь, что для него заданы права на выполнение. Нажмите на нем правой кнопкой мыши, выберите Properties [Свойства], зайдите на вкладку с правами доступа и поставьте галочку Allow executing file as program [Разрешить запускать файл как программу].

Отлично! Вы только что создали простую трехмерную игру для Linux в Unity 5.

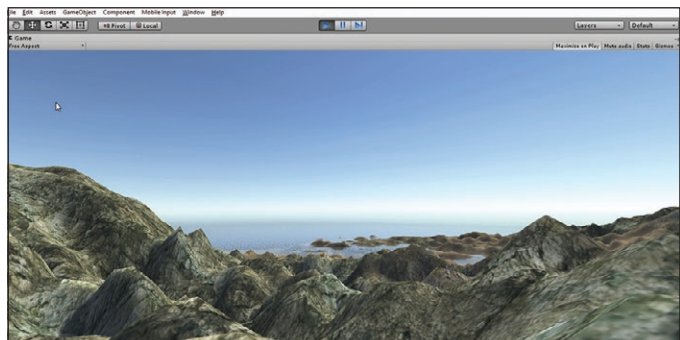
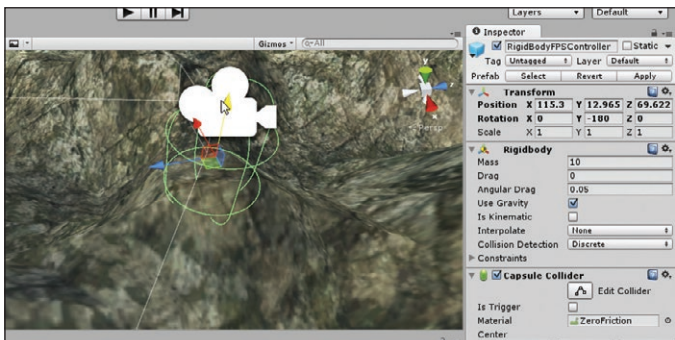


➤ Возможности экспорта для вашей игры обширны, но выбирайте их осмотрительно — это журнал про Linux, и мы знаем, где вы живете.

Скорая помощь

По умолчанию в каждой сцене Unity 5 есть солнце, камера и небо. Попробуйте представлять себе сцену как реальный мир. Не будь у вас неба, солнечного света и глаз, мир был бы темн.

Добавляем игрока

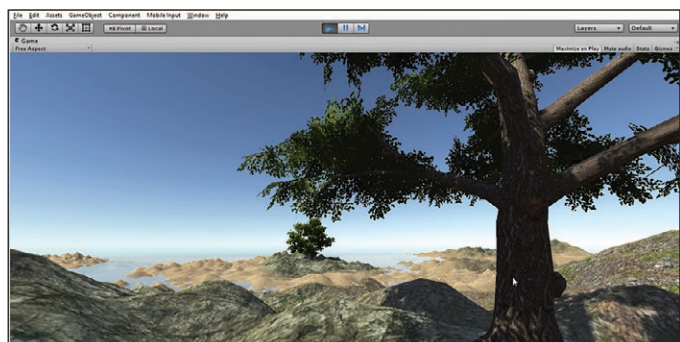
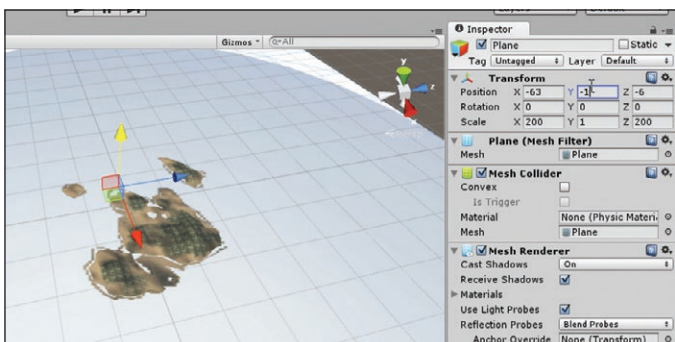


1 Добавим персонаж игрока

Сцена без игрока не слишком интерактивна; давайте добавим Player Controller [Контроллер игрока], чтобы мы могли пройти по острову. На вкладке Project [Проект] выберите Assets > Standard Assets > Characters > FirstPersonCharacter > Prefabs [Ресурсы > Стандартные ресурсы > Персонажи > Персонаж от первого лица > Готовые конструкции] и перетащите контроллер RigidBody на свою поверхность. Приблизьте поверхность и, нажав кнопку Move, тяните за желтые оси до тех пор, пока зеленая ограничивающая фигура не оторвется от земли.

2 Погуляем по острову

Теперь хорошо бы устроить нашей игровой сцене небольшой тест-драйв! Перейдите на вкладку Game [Игра] и нажмите Play [Воспроизвести]. В случае, если вы вдруг обнаружите, что ваш персонаж проваливается под пол, вернитесь к сцене, поднимите игрока чуть повыше и попробуйте снова. После нажатия Play вы сможете поиграть в Робинзона Крузо — прогуляться по своему необитаемому острову и наслаждаться видами пустынных окрестностей. Пока здесь встречаются только скалы да песок.

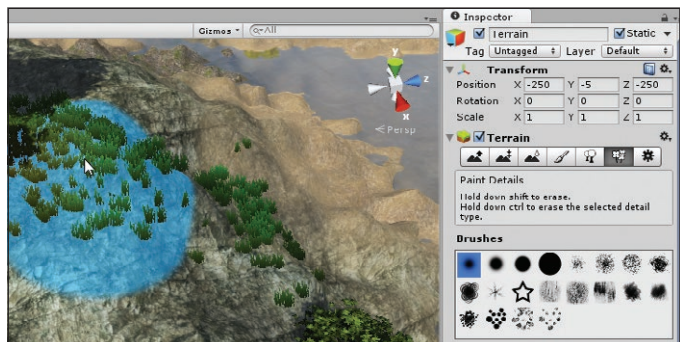
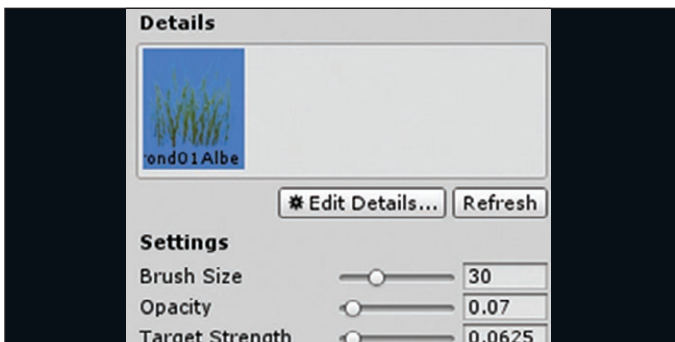


3 Спасение утопающих

Возможно, вы заметили, что игрок может шагнуть в воду и исчезнуть под ней. Добавьте еще один трехмерный объект, на этот раз выбрав Plane [Плоскость]. Задайте для него 200 по осям X и Z, затем установите Transform of Y [Преобразование Y] в -1, опустив эту плоскость под поверхность воды. Поставьте галочку Mesh Collider [Коллайдер сетки] и снимите галочку Mesh Renderer [Рендерер сетки]. Мы создали невидимую плоскость под водой с коллайдером, который препятствует попаданию объектов под воду.

4 Лесной покров для персонажа

Добавим в сцену несколько природных объектов — красивых деревьев. В разделе Terrain [Поверхность] проводника [Explore] выберите Place Trees [Разместить деревья] и нажмите Edit Trees > Add Tree [Изменить деревья > Добавить дерево]. Нажмите на маленький кружок в диалоговом окне Add Tree. Выберите Broadleaf Desktop [Настольное широколиственное дерево] и нажмите Add. Установите Brush Size [Размер кисти] в 20, а Tree Density [Густота леса] в 50. Кистью нарисуйте деревья там, где сочтете нужным.



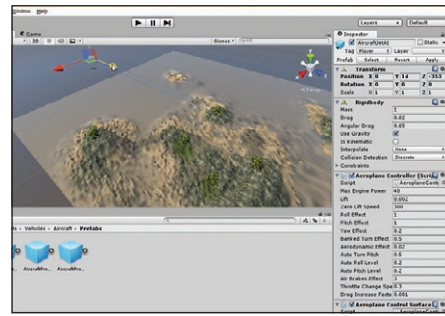
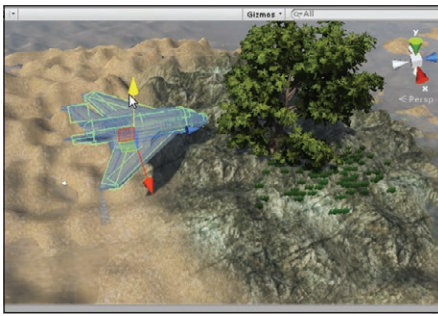
5 Зарастить травой

Перейдем к кнопке рядом под названием Paint Details [Подробные настройки кисти]. Нажмите Edit Details [Изменить настройки] и нажмите Add Grass Texture [Добавить текстуру травы]. Нажмите на кружок и дважды кликните по текстуре GrassFront. Выберите Healthy Color [Здоровый цвет] и сделайте его чуть менее ярким. Сделайте то же самое с сухим цветом [Dry Color]. Нажмите Add. Смените Brush Size, Прозрачность и Нажим [Target Strength] на 30, 0.07 и 0.07 соответственно.

6 Добавим еще травы и соломы

Одного пучка травы на весь остров явно мало. Воспользуйтесь кистями как описывалось прежде и рассадите траву везде, где вам захочется. Воспроизведите свою игру и посмотрите, как она выглядит, затем вернитесь обратно и добавьте еще травы или удалите лишнее, нажав Shift при рисовании. Помните, что трава будет показываться, когда вы находитесь рядом с ней, поэтому при рисовании необходимо увеличивать масштаб.

Добавляем самолет



1 Добавим самолет

Добавим к нашей сцене самолет противника — пусть охотится за нашим героем. Самолет будет пикировать на игрока (как добавить игрока — см. стр. 86), чтобы наш игрок побегал, а не просто любовался пейзажем. Снова перейдите к Standard Assets и найдите Aircraft [Самолеты]. Перетащите AircraftJetAI [Реактивный самолет с искусственным интеллектом] или AircraftPropellerAI [Пропеллерный самолет с ИИ] на вкладку Hierarchy [Иерархия]. Дважды щелкните по центру, чтобы его просмотреть.

2 Настроим искусственный интеллект

Для перемещения самолета воспользуйтесь кнопкой Translate [Перемещение] — это вторая кнопка слева сверху. Мы хотим поместить его над водой, и подальше над морем. В Inspector прокрутите до Aeroplane AI Control [Управление искусственным интеллектом самолета] и выберите кружок, нажав Target, затем выберите RigidBodyFPSController. Вернувшись в инспектор, снимите галочку Waypoint Progress Tracker [Слежение за трекром].

3 Установим барьер для самолета

Пора попробовать игру в действии. Самолет начнет преследовать вас и, скорее всего, врежется в землю. Чтобы решить проблему, выберите невидимый самолет и нажмите правую кнопку мыши, затем выберите Duplicate [Создать дубликат]. Переместите копию в Y 35 или в другое подходящее место для вашей поверхности. Временно включите Mesh Renderer [Рендерер сетки], чтобы увидеть самолет и убедиться, что он не задевает вершин вашей поверхности.

» Очистите строку поиска и выберите Directional Light [Направленный свет] в списке Hierarchy [Иерархия]. Вы увидите, как информация изменится согласно выбранному вами типу освещения.

Смотреть пока нечего, поэтому начнем с воды, так как она придаст визуальный интерес и некие естественные границы миру вашей игры. Под окном сцены расположен список проекта, со всеми ресурсами игры, которые вы загрузили в проект. Выберите Standard Assets > Environment > Water > Water > Prefabs [Стандартные ресурсы > Окружающая среда > Вода > Вода > Готовые конструкции], затем перетащите WaterProDaytime в список иерархии, и она появится в вашей сцене. Слева над сценой вы увидите пять кнопок. Выберите четвертую, нажмите на маленький квадратик в центре элемента Water и, удерживая левую кнопку мыши, переместите мышь вверх. Вода «растянется»

вверх. Вода, которую вы добавили к сцене — это плоский диск, к которому уже применен материал (подробнее о материалах см. врезку внизу). Нажмите кнопку воспроизведения в центре вашего окна сверху, и вы увидите, что здесь еще предлагается. Замечили, что при нажатии кнопки воспроизведения Unity запускает скрипт, иллюстрирующий бег волн? Итак, на данном этапе у нас есть камера — статичная, направленная на горизонт сцен с одним источником света; небо, которое имитирует рассеянный свет, и плоскость с материалом воды; а скрипт создает волны на поверхности воды. Выберите File > Save Scene [Файл > Сохранить сцену] и наберите IslandRun_Scene, затем нажмите Save [Сохранить]. Следуйте приведенным пошаговым инструкциям, чтобы обогатить остров объектами, задать начальную позицию игрока и подослать вражеский самолет. Удачи! LXF

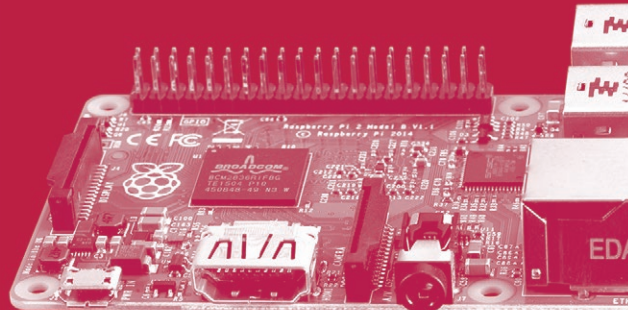
Материалы

С годами игровая графика получила колоссальное развитие, но базовый формат остался прежним. Объекты состоят из сеток [mesh], а те, в свою очередь, из многоугольников-полигонов, обычно треугольников или квадратов. Представьте себе это как сито, которому придали должную форму, облепив его папье-маше. Всё, что вам дадено — трехмерная форма, но без цвета и материалов.

Материалы состоят из текстур, которая в Unity 5 называется Albedo. Она содержит базовую информацию о цвете. К ней прилагаются дополнительные компоненты, такие как карта нормалей [Normal map], определяющая информацию о канавах, трещинах, буграх и царапинах. Материал также может иметь карту поглощения окружающей среды [ambient occlusion map], которая сообщает Unity, где находятся области постоянной тени, или зеркальную карту [specular map], которая сообщает, какие части материала выглядят блестящими. В таких материалах, как наша поверхность, используется карта высот [height map], где высота определяется уровнем шкалы серого в изображении. Для полной раскладки можно посмотреть пример <http://bit.ly/Unity5StandardShader> или более сложный <http://bit.ly/Unity5PhysicallyBasedShading>.



» В Unity 5 есть демонстрационная сцена, помогающая понять, как устроены материалы и освещение.



Получите свою порцию вкуснейших новостей, обзоров и руководств по Raspberry Pi.

ЛИЗ АПТОН
рассказывает
о последних
новостях от фонда
Raspberry Pi.

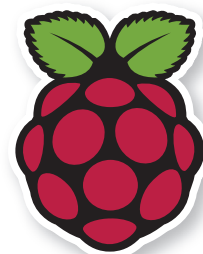


**Добро
ПОЖАЛОВАТЬ...**

Мы в Башнях Pi уже не один месяц работаем не покладая рук: с выхода Pi 2 в начале февраля каждый день продается невероятное количество устройств. Сейчас мы выпускаем 20 000 в день, то есть конвейер работает с должной производительностью, но задел на складе никак не создат. Мы можем оценить цифры только по неоплаченным счетам, но похоже, что на конец марта было продано около 600 000, и нам все равно не удастся запастись ими, чтобы они были в наличии. Срок ожидания поставки сейчас около двух недель, но за эти две недели появляются покупатели еще на две недели!

Эбен [Аптон, Eben Upton] и я за последний месяц побывали в США, Гонконге и Китае, распространяя новости о Pi (а вместе с Мэтом Ричардсоном [Matt Richardson] и Рэйчел Рэйнс [Rachel Rayns] густо раскрасились проводящей краской, чтобы поведать о мире вычислений 500 детям в Техасе). Мы провели вечеринку по случаю трехлетия Raspberry Pi [см. LXF197], где 1300 из вас ели пиццу, посещали семинары и мастер-классы и показывали нам своих роботов/грызунов/говорящую мебель. И будем работать еще активнее, гораздо больше сделав для проекта Astro Pi, чтобы дети смогли отправить свой код на Pi в космос; для наших проектов со школьными метеостанциями и Digital Creatives; и многого другого. Трудно было не только нам. Ребята из *Linux Format* корпели над новым разделом о Raspberry Pi — мы рады его видеть. Думаем, вы найдете массу применений своим зажимам-крокодилам!

Picademy @ Google в Лидсе

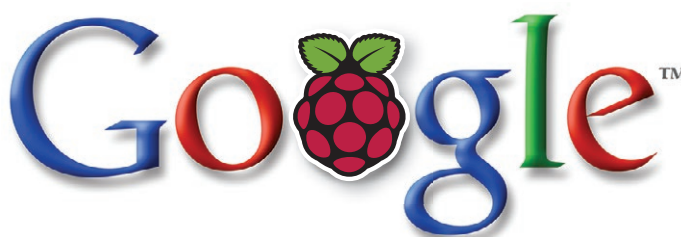


Picademy отправляется на холодный север, чтобы и там наготовить преподавателей.

Фонд Raspberry Pi уделяет массу времени и усилий подготовке преподавателей, которые должны познакомить студентов с радостями программирования на Raspberry Pi, но даже всех проведенных мероприятий Picademy и многих часов, проведенных в дороге, ради посещения конференций, Фонду мало. С щедрой руки Google, которая разрешила Фонду воспользоваться своим «Цифровым гаражом [Google Digital Garage]» в Лидсе, был подготовлен курс Picademy@Google. Он будет проведен специально отобранными членами сообщества и преподавателями и предложит курсы для преподавателей

начальной, средней и старшей школы в сочетании с набором практических занятий, проектов для обучения и общих вопросов.

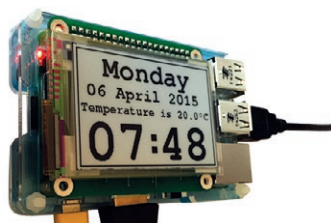
Курсы рассчитаны на два дня и проходили одновременно с классическим курсом Picademy в Pi Foundation с 21 мая. Подробную информацию об этих бесплатных курсах ищите на сайте www.raspberrypi.org/picademy-at-google.



Электронная бумага Pi

Электронная бумага родом не из Йоркшира, разве вы не знали?

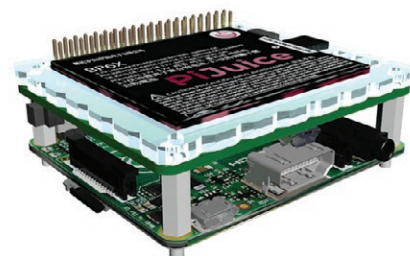
Теперь, благодаря новой кампании на Kickstarter, которая собрала заявленные £10 000 всего лишь за пять дней, Raspberry Pi может управлять даже дисплеем, работающим по технологии электронной бумаги. Новый дисплей соответствует спецификации HAT Pi и содержит работающее от батареи часы реального времени, которыми легко управлять из кода на Python с помощью низкоуровневого драйвера C, который есть на GitHub. <http://bit.ly/pipaper>

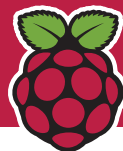


PiJuice

Для Pi появился аккумулятор с возможностью зарядки от солнечных батарей.

Этот умный проект на Kickstarter, который уже полностью собрал все необходимые средства, предлагает портативный и обладающий большой емкостью литий-ионный аккумулятор для питания вашей Pi. Аккумулятор можно даже заряжать от солнечных батарей, что окажется полезным, даже если там, где вы живете, солнце постоянно скрыто за облаками. Это удобный и очень нужный аксессуар, вполне окупающий свою цену — менее £30. <http://bit.ly/pijuice>





Explorer HAT Pro

Наш отважный исследователь **Лес Паундер** надевает Красную Шляпу, берет свой верный хлыст и выезжает искать славы и счастья. И роботов.

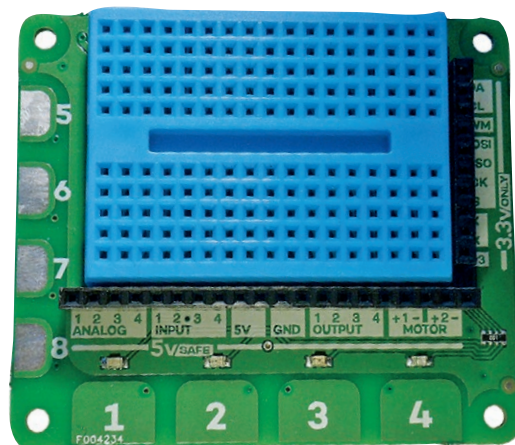
Вкратце

» Последняя HAT-совместимая плата расширения от «Пиратов Шеффилда» представляет собой решение «все в одном» для создания массы различных проектов, от светододов до роботов и всего, что находится между ними. Совместимая с Pi A+/B+ и Raspberry Pi 2, эта универсальная рабочая лошадка доставит много часов удовольствия пользователям любых уровней подготовки.

Pimoroni всегда будут связывать с Raspberry Pi благодаря нескончаемому потоку плат расширения и аксессуаров ее изделия. Начав с многослойного корпуса Pirow в 2012 г., команда Pimoroni постепенно набиралась опыта, и ее последний продукт, Explorer HAT Pro, лишнее подтверждение их достижений.

Explorer HAT Pro обладает почти всем, чего можно желать от платы расширения: емкостные сенсорные точки, внешние сенсорные входы в стиле Makey Makey [комплект для превращения разных предметов в сенсорные кнопки], аналоговый вход, набор цифровых входов/выходов, четыре светодиода с поверхностным монтажом, дополнительная плата и контроллеры двигателя. Это много плат за умеренные деньги. Как вы, вероятно, уже догадались, это плата стандарта HAT, совместимая с новыми моделями Raspberry Pi 2 и моделями A+ и B+ исходной Pi. Установка относительно проста: плату нужно подключить ко входам/выходам общего назначения и установить программу удобным скриптом *Bash* из терминала.

На плате Explorer HAT Pro есть 8 емкостных сенсорных точек: точки 1–4 используются как есть, для их активации достаточно легкого касания. Точки 5–8 можно использовать как дополнительные сенсорные входы, но лучше соединить их зажимами «крокодил» с проводящими предметами, такими как фрукты или станиоль. Это позволяет реализовать управление в стиле Makey Makey, но с обширной библиотекой на Python для интеграции в ваши собственные проекты на Python. При



» Explorer HAT Pro подойдет к 40-контактной плате GPIO Pi A+, B+ и Raspberry Pi 2. Благодаря своим размерам она очень точно разместится на A+.

помощи аналоговых входов Explorer HAT Pro удобно считывать сигналы с аналоговых компонентов, таких как фоторезисторы. На Raspberry Pi нет аналоговых входов/выходов, поэтому нужен некий аналог АЦП. С помощью светодиодов удобно реализовать веселые проекты, такие как «светодод», и придать изюминку вашим проектам. Explorer HAT Pro имеет серию защищенных контактов 5 В, которые предоставляют уровень защиты между Raspberry Pi и токами, потенциально способными нанести ущерб схеме.

Двигатели

Заметное преимущество Explorer HAT Pro — контроллер двигателей. Он предоставляет два выхода для управления двигателями с полным реверсивным управлением благодаря H-мосту. Это позволяет вращать робота на все 360°, изменять направление движения робота и полностью управлять им — то, что было невозможно на более ранней плате расширения Pi-brella. Для реализации всех этих прекрасных возможностей поверх платы устанавливается макетная плата с контактами, на которую можно установить распространенные компоненты и интегрировать их в свои проекты.

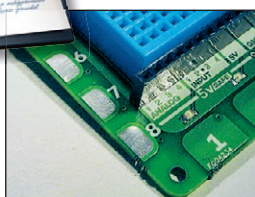
Для поддержки «железа» в Explorer HAT Pro есть обширная библиотека Python, которая работает с Python 2 и 3. Синтаксис библиотеки очень напоминает более ранний проект Pi-brella, который разрабатывался совместно Pimoroni и Syntech, и это существенно понижает

«планку входа». Если вам случалось пользоваться платой Pi-brella раньше, то вы уже наполовину знаете, как работать с Explorer HAT Pro. Полное руководство по Explorer HAT Pro есть в репозитории Github Pimoroni (<https://github.com/pimoroni/explorer-hat>).

На момент написания статьи Explorer HAT Pro поддерживается в версии Scratch-GPIO 7 для разработчика, что позволяет полностью использовать его в Scratch-GPIO, а значит, дети быстрее получат удовольствие от создания схем.

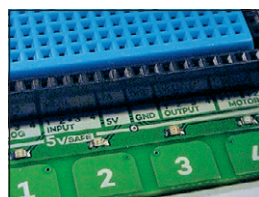
Explorer HAT Pro — самая богатая возможностями плата этой серии. У нее есть более дешевый аналог, Explorer HAT, но без аналоговых входов и функций управления двигателем. И гораздо эффективнее купить Pro-версию и наслаждаться «полноценной» версией платы. **LXF**

Свойства навскидку



Сенсорный интерфейс

Восемь точек касания позволят добиться интерактивности, способной конкурировать с популярной продукцией от Makey Makey.



Рулим двигателями

Разработать роботов и другие моторизованные создания помогут простая библиотека на Python и точное управление 5-вольтовыми двигателями.

LINUX FORMAT Вердикт

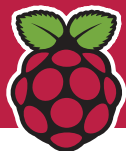
Explorer HAT Pro

Разработчик: Pimoroni
Сайт: pimoroni.com
Цена: £18

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

» Простое ПО и обширные аппаратные возможности позволяют легко воспользоваться этой платой в самых разнообразных проектах.

Рейтинг 8/10



GPIO Python: Контроль дороги

Лес Паундер учит регулировать уличное движение, чтобы пешеходы безопасно переходили дорогу по светофору под управлением Raspberry Pi.



Наш эксперт

Лес Паундер — член образовательной команды Raspberry Pi Foundation и ведет учебные курсы в Picademy. Ему нравится (разумеется) Pi, а также учиться создавать новые устройства.

Существует масса систем, которые можно смоделировать с помощью программ: ядерные реакторы, погодные особенности и самое простое — скромный светофор. Светофор — реальная физическая система, которую легко имитировать с помощью Raspberry Pi и кода на Python. Для этого проекта подойдет любая модель Raspberry Pi; в данном случае мы воспользуемся Raspberry Pi 2, также понадобятся макетная плата, 3 светодиода LED (Light emitting diode) — зеленый, красный и желтый, три резистора по 220 Ом, один переключатель с самовозвратом, зуммер и два набора соединительных проводов Jumper Jerkys (один «мама-папа» и один наоборот). Мы воспользовались прекрасным EduKit от CamJam, который стоит примерно £5 — его можно купить на The Pi Hut: <http://thepihut.com/collections/camjam-edukit>.

Начнем с подготовки нашего Raspberry Pi к использованию. Ваша задача — подключить компоненты к Raspberry Pi на макете. Схема расположения компонентов в высоком разрешении доступна на <http://bit.ly/LXF198Beginner>.

Когда все будет готово, загрузите рабочий стол. Для этого проекта нам потребуется приложение Python 3, которое есть в меню Programming, но поскольку мы будем пользоваться входами/выходами общего назначения (GPIO), приложение нужно открыть особым образом, воспользовавшись терминалом и запустив команду с префиксом **sudo**. Посмотрев в левый верхний угол экрана, вы увидите иконку с черным монитором; это *LXTerminal*. Щелкните по ней один раз левой кнопкой мыши, и откроется окно терминала. В терминале введите следующее, не забыв нажать Enter в конце:

```
sudo idle3
```

Эта команда откроет редактор Python 3, известный как IDLE (Integrated DeveLopment Environment — интегрированная среда разработки), где мы и будем программировать наш проект. Перед тем, как начать, проработаем логику с помощью псевдокода:

```
Для начала зажжем зеленый LED, наш "трафик" разрешен
Ждем нажатия кнопки [button] пользователем
Кнопка нажата
  Выключить зеленый [green] LED
  Включить желтый [amber] LED
  Ждать 2 секунды
  Выключить желтый LED
  Включить красный [red] LED
  10 раз
  Включить зеленый LED и зуммер [buzzer]
  Ждать 0.2 секунды
  Выключить зеленый LED и зуммер
  Включить желтый LED
  Ждать 2 секунды
  Выключить красный и желтый LED
Вернуться на начало
```

Приступим к программированию. В IDLE выберите File > New [Файл > Создать], чтобы открыть новое окно редактора. В нескольких первых строках кода мы импортируем библиотеки, которые помогут вам воспользоваться контактами GPIO на Raspberry Pi и управлять задержками в последовательности зажигания светодиодов.

```
import RPi.GPIO as GPIO
from time import sleep
```

Потом настроим GPIO, сначала попросим их не быть слишком подробными с сообщениями об ошибках, а затем сказав Python, что мы будем пользоваться обозначениями контактов Broadcom.

```
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
```

Чтобы упростить работу с обозначениями контактов GPIO, воспользуемся пятью переменными, которые будут «именами» для соответствующих контактов GPIO.

```
red = 14
amber = 15
green = 18
button = 23
buzzer = 25
```

Теперь вы знаете, какие контакты будут использоваться для светодиодов, зуммера и кнопки, и нужно объяснить Python, как работать с каждым из них. Примечание: светодиодам и зуммеру будут соответствовать выходы, а кнопке — вход.

```
GPIO.setup(red, GPIO.OUT)
GPIO.setup(amber, GPIO.OUT)
GPIO.setup(green, GPIO.OUT)
GPIO.setup(buzzer, GPIO.OUT)
GPIO.setup(button, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

Управление светофором

Теперь перейдем к основной части кода, которая управляет последовательностью работы светофора. Начнем с **try**, который является частью блока обработки ошибок **try...except**. Внутри этого блока добавим бесконечный цикл, **while True**. Затем включим зеленый светодиод и выведем сообщение в оболочку Python, а затем выполним задержку в 0.2 с:

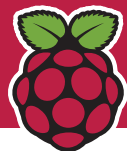
```
try:
  while True:
    GPIO.output(green,1)
    print("Зеленый свет, можно двигаться")
    sleep(0.2)
```

Следующий фрагмент кода обрабатывает нажатие кнопки, и так как фрагмент довольно большой, мы разделим его на части. Во-первых, если кнопка нажата, проверка вернет False, потому что в начале кода мы зарегистрировали состояние кнопки как True с помощью вызова **GPIO.setup**. Мы установили входной контакт в **High (True)**, поэтому при нажатии кнопки его состояние



Скорая помощь

Существует две схемы обозначения контактов GPIO Pi: BCM и Board. Функция **GPIO.setmode()** сообщает Pi, какую схему выбрали вы для своего проекта, и нужно придерживаться ее.



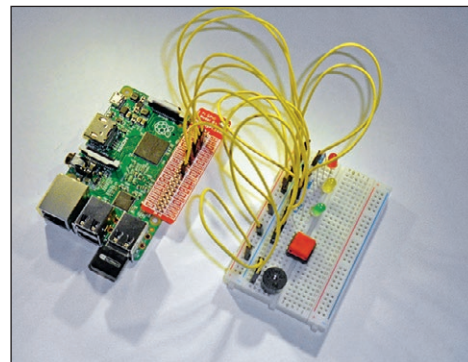
Контакты GPIO

Что же делает Raspberry Pi такой расширяемой платформой для изобретательства? Набор контактов, образующих GPIO — входы/выходы общего назначения [General Purpose Input Outputs]. В оригинальном Raspberry Pi для использования было доступно всего 26 контактов, но с выпуском B+ в середине 2014 г. это количество возросло до 40, и теперь по умолчанию во всех моделях Raspberry Pi их ровно столько.

GPIO используются для подключения электронных компонентов, таких как светодиоды, резисторы и зуммеры, которыми затем можно управлять с помощью Scratch GPIO или, в нашем случае, Python. GPIO также можно использовать для подключения к устройствам I2C (Inter-Integrated Circuit), таким как проект Flotilla от Pimoroni, который скоро выйдет. В I2C для управления многими устройствами, подключенными к шине,

используются всего четыре провода. GPIO также поддерживает SPI, ближний синхронный последовательный интерфейс, который часто используется с SD-картами.

Во многих проектах с использованием GPIO вам потребуется воспользоваться платами расширения HAT (Hardware Attached on Top — устройство поверх), или макетами. Если вы собираетесь заняться любыми проектами с электронными компонентами, советуем купить макетную плату. Макетная плата — это стандартный электронный аксессуар, идеально подходящий для создания прототипов электрических схем и прочего. Макетные платы позволяют удобно и динамично соединять компоненты и провода с помощью решетки промежуточных соединений. Макетные платы дешевы, и они есть во всех хороших магазинах электронных товаров.



► Подключаться к GPIO можно при включенном Raspberry Pi; однако убедитесь, что всё делаете правильно, так как короткое замыкание может вызвать перезагрузку Pi.

изменяется на **Low (False)**, и это вызывает отключение зеленого светодиода, после чего на две секунды включается желтый и затем отключается и, наконец, загорается красный и движение останавливается.

```
if GPIO.input(button) == False:
    GPIO.output(green,0)
    GPIO.output(amber,1)
    sleep(2)
    GPIO.output(amber,0)
    GPIO.output(red,1)
```

Защипываем светодиоды

Во втором фрагменте кода используется цикл **for** с 10 повторениями, он отвечает за мигание зеленого светодиода и гудение зуммера, быстро включая и выключая их. После этого мы включаем красный и желтый светодиод, ждем две секунды и затем выключаем их. Весь этот цикл повторяется до тех пор, пока пользователь не нажмет кнопку для перехода через перекресток. Если этого не произойдет, будет гореть зеленый, чтобы движение продолжалось.

```
for i in range(10):
    print(" W A L K ")
    GPIO.output(green,1)
    GPIO.output(buzzer,1)
    sleep(0.2)
    GPIO.output(green,0)
    GPIO.output(buzzer,0)
    sleep(0.2)
    GPIO.output(red,1)
    GPIO.output(amber,1)
    sleep(2)
    GPIO.output(amber,0)
    GPIO.output(red,0)
```

Наконец, закроем конструкцию **try...except**, чтобы она приняла **KeyboardInterrupt**. Обычно это делается нажатием **Ctrl+C** для остановки запущенной программы. При возникновении данного события программа вернет состояния всех GPIO в состояния по умолчанию и затем корректно завершится.

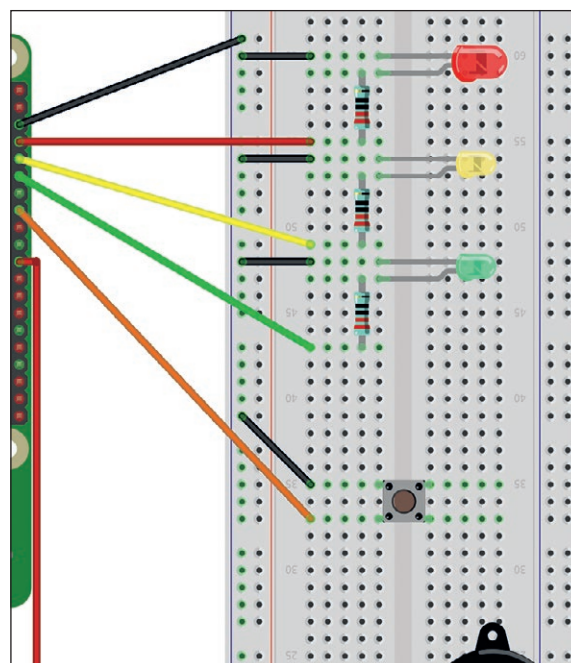
```
except KeyboardInterrupt:
    GPIO.cleanup()
```

Написав код, его нужно сохранить, и после этого выберите **Run > Run Module** [Запуск > Запустить модуль]. Подождите несколько секунд, и должен загореться зеленый светодиод. Теперь нажмите

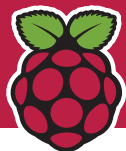
кнопку, и должен загореться желтый светодиод и затем красный. Зеленый светодиод начнет мигать, и зуммер подаст сигнал о переходе проезжей части.

Поздравляем! Вы смоделировали первую физическую систему с помощью небольшого количества кода на Python, нескольких электронных компонентов и Raspberry Pi.

В нашем проекте мы воспользовались необходимыми компонентами из набора EduKit #1 от CamJam. На момент написания статьи предлагалось два варианта наборов. EduKit #1 — тот, которым мы воспользовались — содержит основные компоненты для различных проектов, связанных со входами/выходами, и хотя набор EduKit #2 был выпущен в конце 2014 г., он дополнил первый набор датчиками, такими как PIR (Passive Infra Red — пассивный инфракрасный датчик) и датчики температуры. В оба набора входят схемы, которые скоро перейдут под управление образовательной командой Raspberry Pi Foundation. Оба набора можно купить на <http://thePIHut.com/collections/camjam-edukit>; первый продается всего за £5, а второй чуть дороже — за £7, но он вполне оправдывает свою стоимость. **LXF**



► Эта схема сделана в великолепной программе под названием **Fritzing** (<http://fritzing.org/home>). В ней легко разрабатывать проекты и делиться ими с другими.



Raspbian: Строим сервер на Pi 2

Лес Паундер применяет новенький Raspberry Pi 2 в качестве небольшого, но энергоэффективного сервера для своей растущей цифровой библиотеки.



Наш эксперт

Лес Паундер специализируется в обучении британских преподавателей Raspberry Pi. Он ведет блог на <http://bigl.es>.

Традиционно серверы — это огромные устройства, которые потребляют немало энергии и выделяют значительное количество тепла, и поэтому дата-центры представляют собой большие кондиционируемые помещения, где полным-полно стоек, в каждой из которых по несколько серверов. Но для домашнего использования и для небольших предприятий это уже не обязательно так, и с появлением Raspberry Pi мы получили компактный компьютер с Linux, потребляющий очень мало энергии и выделяющий малое количество тепла. На нашем уроке мы употребим Raspberry Pi 2 для построения небольшого, мощного и энергоэффективного файлового сервера, к которому можно будет подключаться по SSH для удобного размещения резервных копий.

Кроме Pi, вам понадобятся чистая карта microSD, хороший источник питания для Raspberry Pi, Ethernet-подключение к роутеру и внешний жесткий диск USB с внешним питанием. Начнем с создания SD-карты, содержащей операционную систему. С технической точки зрения, это набор программ на основе ядра Linux, который называется «дистрибутивом». Мы выберем дистрибутив Raspbian, который по умолчанию используется для Raspberry Pi.

Существует два способа установки Raspbian на SD-карту. Первый способ (NOOBS) требует, чтобы вы загрузили ZIP-архив NOOBS и затем распаковали его содержимое на чистую карту, отформатированную в FAT. Вставьте эту карту в Raspberry Pi, и системное меню спросит вас о том, какой дистрибутив устанавливать; в меню надо выбрать Raspbian, а затем попить чайку, пока он устанавливается. Это простейший способ установки Raspbian, но чтобы от установки осталось меньше следов, лучше воспользоваться другим способом — загрузить образ Raspbian и распаковать его из архива, скопировав его на чистую SD-карту командой `dd`.

Команда `dd` очень мощная, но опасная. При неправильном использовании она способна уничтожить все данные на диске, поэтому пользуйтесь ею с осторожностью. Прежде чем взять на вооружение `dd`, нужно идентифицировать SD-карту, которая

вставлена в компьютер. Откройте *LXTerminal* (щелкнув по иконке на рабочем столе) и наберите следующее:

```
sudo fdisk -l
```

Теперь вы сможете увидеть список томов/дисков, которые найдет `fdisk`. Обычно все, что начинается с `'/dev/sdX'` — это внутренние жесткие диски компьютера. Пройгнорируйте их и ищите нечто вроде `'/dev/mmcblk0XX'`, где вместо XX может быть p1 или p2. P1/p2 можно проигнорировать, поскольку это разделы на карте, а мы будем записывать данные на всю карту, а не на конкретный раздел. Если вы видите такую запись, это означает, что карта идентифицирована. Запомните ее местоположение.

SD-карта обнаружена; скопируем образ Raspbian на чистую карту командой `dd`. В *LXTerminal* перейдите в каталог, куда вы распаковали образ Raspbian из ZIP-архива; обычно это `Downloads`.

```
cd ~/Downloads
```

Теперь командуйте `dd`. Пожалуйста, перед нажатием Enter всё дважды проверьте, так как после этого остановить процесс не получится. В *LXTerminal* выполните следующую команду, заменив `raspbian.img` и `/dev/mmcblk0` на соответствующие образ и устройство для своей системы, полученные с помощью команды `fdisk -l`.

```
sudo dd if=./raspbian.img of=/dev/mmcblk0 bs=4M
```

У вас запросят пароль, и затем `dd` приступит к работе, но в процессе ее деятельности вы ничего не увидите — команда просто сделает свою работу и сообщит об этом через 5–10 минут. Так что у вас есть время выпить чашечку чая.

Настройка Rasp-config

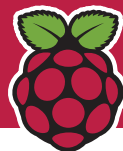
Подготовив SD-карту, приступайте к сборке Raspberry Pi. Для начальной настройки Pi нужно подключить к ней монитор, клавиатуру и мышь. Сделав это, вставьте SD-карту и включите питание Pi. Загрузка Pi 2 займет не более 15 секунд, и по ее окончании откроется окно *raspi-config*. Ваша первая задача — расширить файловую систему на SD-карте, чтобы получить максимальное количество пространства. Для этого перейдите к пункту 1 и нажмите Enter, чтобы запустить автоматический процесс. Сделав



➤ SSH — защищенный протокол, который шифрует ваше соединение, предоставляя вам более безопасный способ передачи данных через Интернет.

```
A1 Overscan      You may need to configure overscan if black bars are present on display
A2 Hostname      Set the visible name for this Pi on a network
A3 Memory Split  Change the amount of memory made available to the GPU
A4 SSH           Enable/Disable remote command line access to your Pi using SSH
A5 Device Tree   Enable/Disable the use of Device Tree
A6 SPI           Enable/Disable automatic loading of SPI kernel module (needed for e.g. PiFace)
A7 I2C           Enable/Disable automatic loading of I2C kernel module
A8 Serial        Enable/Disable shell and kernel messages on the serial connection
A9 Audio         Force audio out through HDMI or 3.5mm jack
A0 Update        Update this tool to the latest version

<Select>                                <Back>
```



Создаем сетевой принтер

Обычно в домах при нескольких компьютерах бывает всего один принтер, к которому приходится подключаться, чтобы напечатать документ. А не создать ли общий сетевой принтер с помощью нашего сервера на Pi? Ну, это возможно. Для этого воспользуемся CUPS, Common Unix Printing System [Общая система печати Unix], сервисом печати, созданным Apple и применяемым в системах OS X и Unix по всему миру.

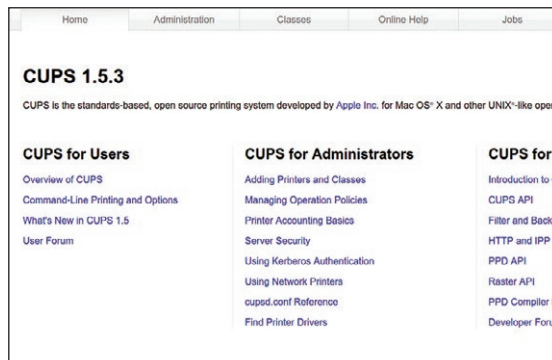
Обычно мы запускаем CUPS на своем компьютере, но также можем попросить наш компьютер напечатать документ на сетевом принтере. Для установки CUPS надо подключиться по SSH к своему Pi и набрать в *LXTerminal* следующее:

```
sudo apt-get install cups
```

Затем надо внести изменения в настройки. Мы нашли прекрасное руководство на *How-To Geek* (<http://bit.ly/LFX198-cups>), иллюстрирующее выполнение этой задачи на Raspberry Pi. Все, что нужно сделать для администрирования удаленного принтера CUPS — открыть браузер и набрать адрес своего сервера Pi, добавив к нему **:631**:

```
192.168.0.6:631
```

Здесь можно добавить новые принтеры, проверить состояние задач печати и отменить любые ошибочные задачи. В крайнем случае можно даже настроить удаленное задание на печать, воспользовавшись перенаправлением портов на своем роутере для перенаправления заданий печати извне сети на свой принтер CUPS.



» В CUPS есть удобный административный web-интерфейс, который позволяет удаленно управлять принтерами и добавлять новые принтеры с помощью мастера.

это, перейдите к пункту 8 — Advanced options [Расширенные параметры]. Перейдите к A3, Memory Split [Распределение памяти] и нажмите Enter. Измените значение на 16 и нажмите Enter. При этом GPU будет выделено 16 МБ оперативной памяти. Так как Pi будет использоваться без монитора, то большого количества видеопамати не требуется.

После этого вы перейдете в главное меню, но следует вернуться обратно в меню Advanced options — пункт 8. В этом меню перейдите к A4, SSH, переместите курсор в Enable и нажмите Enter. На этом наша настройка закончена. Вернитесь в главное меню, выберите Finish [Завершить] и нажмите Enter. Для завершения настройки перезагрузите Raspberry Pi.

Перезагрузив Pi, снова войдите в систему. Имя пользователя по умолчанию — pi, пароль — raspberrypi. Теперь обновите свой дистрибутив. В *LXTerminal* наберите следующую команду и нажмите Enter:

```
sudo apt-get update
```

Команда сравнит список установленных пакетов с списком на серверах Raspbian и обновит нужные пакеты. Если список для загрузки получится большим, Raspbian спросит вас, надо ли затевать эту работу. В зависимости от скорости интернет-соединения загрузка может занять некоторое время, но по окончании процедуры управление вернется к вам.

Установка фиксированного IP-адреса

По завершении обновлений можно создать фиксированный IP-адрес для сервера Pi. Для этого следует изменить файл `/etc/network/interfaces`. Вам потребуется узнать текущий IP-адрес Pi — его можно определить командой

```
ifconfig
```

Эта команда выведет много текста. Так как у вас Ethernet-соединение, ищите строку eth0, обычно она самая первая. В ней найдите inet addr:. Наша конфигурация выглядит так:

```
eth0 Link encap:Ethernet HWaddr b8:27:eb:b8:d2:c8 inet addr:
192.168.0.6 Bcast:192.168.0.255 Mask:255.255.255.0
```

Чтобы получить более подробную информацию, надо также запустить `netstat`:

```
netstat -nr
```

и найти и записать два адреса — шлюз [gateway] и адрес назначения [destination].

Теперь откройте файл настройки `/etc/network/interfaces`:

```
sudo nano /etc/network/interfaces
```

В конфигурации по умолчанию интерфейс eth0, Ethernet, использует DHCP (Dynamic Host Configuration Protocol — протокол динамической конфигурации хоста). Другими словами, компьютер получает IP-адрес при каждом подключении к роутеру:

```
iface eth0 inet dhcp
```

Чтобы знать, где в сети находится наш сервер Pi, мы зафиксируем его IP-адрес в то значение, которое нашли с помощью `ifconfig`; в нашем случае это 192.168.0.6, но у вас оно будет отличаться. Мы внесем изменения только в раздел `iface eth0`; другие разделы нужно оставить без изменений.

Наш новый раздел выглядит так (а ваш будет отражать настройки вашей сети):

```
iface eth0 inet static
# IP-адрес, который мы хотим использовать
address 192.168.0.6
# Используется, чтобы разбить IP-адрес на подсети, это стандартно и для вас сработает
netmask 255.255.255.0
# Структура IP-адресов нашей сети; ваша может быть 192.168.1.0
network 192.168.0.0
# Мы нашли это через ifconfig ранее, с меткой BCAST
broadcast 192.168.0.255
```

```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:b8:d2:c8
          inet addr:192.168.0.6  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:409864 errors:0 dropped:0 overruns:0 frame:0
          TX packets:655914 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:357102357 (340.5 MiB)  TX bytes:712992155 (679.9 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4204 (4.1 KiB)  TX bytes:4204 (4.1 KiB)

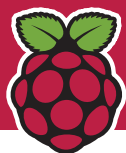
pi@raspberrypi ~ $
```

» Команда `ifconfig` обладает большими возможностями, и при ее помощи можно просматривать настройки интерфейсов и менять их на лету.

Скорая помощь

Чтобы сделать этот проект слегка необычным, можно добавить плату Unicorn HAT (<http://pimoroni.com>) и получить визуальную информацию об использовании процессора, успешном резервном копировании и неисправностях. Все это возможно благодаря скрипту на Python, запускаемому при загрузке.

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!



Скорая помощь



Raspberry Pi В+ и Raspberry Pi 2 Model В работают с внешними USB-дисками лучше своих предшественников. Первоначально подключение USB-диска к Pi вызвало перезагрузку, но начиная с модели В+ эту ошибку исправили.

```
# Адрес шлюза, найденный по netstat -nr
gateway 192.168.0.1
```

Когда будете готовы, сохраните изменения, нажав Ctrl+O, затем выйдите, нажав Ctrl+X. Перезагрузите Raspberry Pi, набрав `sudo reboot`

После перезагрузки вы вернетесь в строку входа в систему. Войдите в систему и наберите

```
ifconfig
```

чтобы убедиться, что IP-адрес теперь стал статическим; если вы хотите проверить наличие доступа к Интернету, наберите следующее:

```
ping google.com
```

Если пинг пройдет удачно, сетевые настройки заданы верно. Теперь можно продолжить работы с клавиатурой и монитором или отсоединить их и подключиться к Pi по сети. Мы предпочли подключиться по SSH с нашего ноутбука с Linux Mint. Если соберетесь сделать так же, откройте терминал на своем компьютере и наберите

```
ssh pi@IP ADDRESS
```

Не забудьте подставить сюда статический IP-адрес, который мы задали ранее. Для входа в систему будет нужно указать пароль, но после этого вы сможете отправить на Raspberry Pi любые команды.

Собрав систему, нужно настроить ее как сервер, и первым делом разрешить Raspbian записывать и считывать данные с нашего внешнего жесткого диска USB 2.0. Если диск отформатирован

в одной из типичных файловых систем Linux — ext3, ext4 или btrfs — им можно пользоваться без дополнительной настройки. Но если диск отформатирован в NTFS — файловой системе Windows, надо установить утилиту, которая позволит записывать и считывать данные с диска, и она называется *NTFS-3G*. Чтобы ее установить, наберите следующую команду в *LXTerminal*:

```
sudo apt-get install ntfs-3g
```

Через несколько секунд, по завершении установки, управление вернется к вам.

Установка Webmin

Администрирование сервера традиционно выполняется в терминале, в данном случае в *LXTerminal*. Вообще-то мы уже это делали. Однако, чтобы немного упростить администрирование сервера, есть великолепная программа под названием *Webmin*. Это веб-интерфейс для распространенных административных задач, таких как управление пользователями и обновление ПО.

Чтобы установить *Webmin* на сервер Pi, воспользуйтесь менеджера пакетов *APT*, который загрузит программу, а заодно и все ее зависимости.

```
sudo apt-get install webmin
```

Установив *Webmin*, откройте в браузере адрес своего Pi, добавив номер порта 10000. Наш IP-адрес выглядел так: <https://192.168.0.6:10000/>.

Может появиться предупреждение, что сертификат не заслуживает доверия. В этом случае не беспокойтесь, просто выберите *Advanced Options* и идите дальше. Появится окно входа в систему — в нем введите логин и пароль, с помощью которых вы ранее входили в Pi.

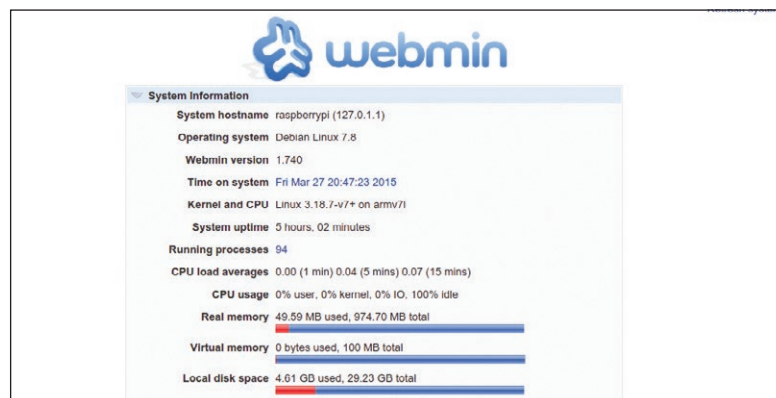
Зайдя в систему, вы увидите главное меню, разбитое на 8 разделов. В этом проекте нам понадобится меню *System* [Система], с помощью которого можно управлять жесткими дисками, а также пользователями и группами.

Следующая задача — сделать так, чтобы внешний USB-диск монтировался при загрузке, поэтому подключите диск к USB-порту Pi. До того, как вы сможете управлять диском в *Webmin*, нужно вернуться в *LXTerminal* и создать каталог, который будет точкой доступа. В терминале зайдите в каталог `/media/` и создайте новый каталог `drive`, следующим образом:

```
cd /media
```

```
sudo mkdir drive
```

После создания каталога у нас появилась проблема: этим каталогом могут воспользоваться только пользователи с правами `sudo` или `root`. Вам нужно изменить права доступа так, чтобы



» *Webmin* предоставляет графический интерфейс для решения распространенных задач администрирования серверов. Через него можно даже обновлять ПО на Raspberry Pi и настраивать задания *Cron*.

Доступ к серверу на Pi

В этом проекте мы создали в своей домашней сети файловый сервер, который недоступен за пределами этой сети. Если сервер надо сделать доступным извне, следует предпринять ряд мер, которые мы свели в удобный список:

- » Измените пароль по умолчанию для пользователя Pi, которому нужно открыть доступ. Это делается командой `passwd` в *LXTerminal*.
- » Создайте нового пользователя без привилегий `sudo` или `root`; проще всего это сделать в меню *System > Users and Groups* [Система > Пользователи и группы].
- » Настройте перенаправление портов на роутере, чтобы запросы на внешний IP-адрес перенаправлялись на сервер Pi. Этот процесс немного отличается для разных роутеров, поэтому гляньте в руководство своего роутера.

» Ваш провайдер, скорее всего, не предоставляет статического внешнего IP-адреса. Если это так, установите *noipclient*, который можно загрузить с <http://bit.ly/No-ipLinux>, и создайте учетную запись на сайте программы (www.noip.com).

Клиентская программа сменил ваш IP-адрес на URL, который вы создаете на сайте. С помощью этого URL вы легко подключитесь к серверу, и указывать его IP-адрес не понадобится.

Наконец, очень важно настроить брандмауэр, который защитит сервер от внешних воздействий. Проще всего установить на сервере *fwbuilder* и *VNC*, так как *fwbuilder* — графическое приложение для настройки правил брандмауэра.

Username	User ID	Group	Real name	Home dir
root	0	root	root	/root
daemon	1	daemon	daemon	/usr/sbin
bin	2	bin	bin	/bin
sys	3	sys	sys	/dev
sync	4	nogroup	sync	/bin
games	5	games	games	/usr/games
man	6	man	man	/var/cache
lp	7	lp	lp	/var/spool
mail	8	mail	mail	/var/mail
news	9	news	news	/var/spool
uucp	10	uucp	uucp	/var/spool
proxy	13	proxy	proxy	/bin
www-data	33	www-data	www-data	/var/www
backup	34	backup	backup	/var/backu
list	38	list	Mailing List Manager	/var/list
irc	39	irc	ircd	/var/run/irc
gnats	41	gnats	Gnats Bug-Reporting System (admin)	/var/lib/gnats
netbsd	65534	nogroup	netbsd	/usr/share
ibmuser	100	ibmuser		/var/lib/ibm
pi	1000	pi		/home/pi

» Через *Webmin* также можно управлять пользователями и группами в графическом интерфейсе. Для новичков в администрировании серверов это гораздо удобнее.



пользователи могли читать и записывать данные на жесткий диск. Для этого наберите в *LXTerminal* следующее:

```
sudo chmod 770 ./drive
```

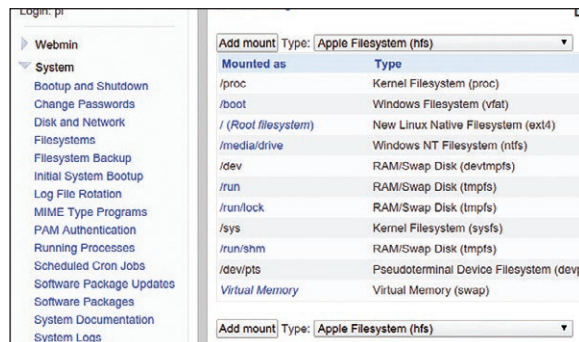
На этом с терминалом пока все. Вернемся в *Webmin*.

Монтирование диска при загрузке

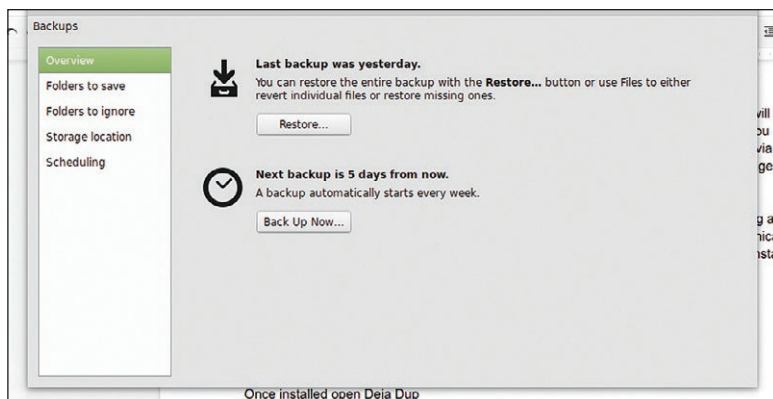
Для управления USB-дисками в *Webmin* выберите меню System в левой части окна, затем нажмите на Disk and Network Filesystems [Дисковая и сетевая файловые системы]. Через несколько секунд на экране появится список всех смонтированных файловых систем. Нашего диска в списке не будет, и нужно его добавить, нажав на кнопку Add Mount [Смонтировать новый диск]; но прежде чем это делать, посмотрите направо, и вы увидите выпадающий список под названием Apple Filesystem (HFS) [Файловая система Apple (HFS)]. Щелкните по нему и выберите Windows NT Filesystem NTFS, если ваш диск отформатирован в NTFS. Если нет, выберите соответствующую своему диску файловую систему. После выбора нажмите кнопку Add Mount, и откроется новое меню.

В новом меню начните с кнопки Mounted As и ... справа от него. Откроется диалоговое окно, в котором можно открыть каталог /media/drive directory, который мы создали ранее. Для остальной настройки воспользуйтесь простым списком:

- » Save Mount? Save and Mount at Boot [Сохранить? Сохранить и монтировать при загрузке].
- » Mount Now? Mount [Смонтировать сейчас? Нет].
- » Windows NT Filesystem: Disk. Look for your device in the dropdown menu [Файловая система Windows NT: Диск. Найдите свой диск в выпадающем меню].
- » Mount Options [Параметры монтирования].
- » Read-only? No [Только для чтения? Нет].
- » Allow users to mount this filesystem? Yes [Разрешить пользователям монтировать эту файловую систему? Да].
- » Disallow execution of binaries? If mountable by users [Запретить исполнение двоичных файлов? Если монтируется пользователями].
- » Avoid updating last access time? No [Отключить обновление времени последнего обращения? Нет].
- » Buffer writes to filesystem? Yes [Буферная запись в файловую систему? Да].
- » Disallow device files? If mountable by users [Отключить файлы устройств? Если монтируется пользователями].
- » Disallow setuid programs? If mountable by users [Отключить программы, изменяющие setuid? Если монтируется пользователями].
- » Wait until network interfaces are up? No [Подождать, пока сетевые интерфейсы загрузятся? Нет].
- » User files are owned by: Leave blank [Владельцем файлов является пользователь: оставить пустым].
- » Group files are owned by: Leave blank [Владельцем файлов является группа: оставить пустым].



» Создание новой точки монтирования бывает сложным для новичков, но с *Webmin* этот процесс упрощается. Главное, о чем следует помнить — кнопка Add Mount: прежде чем на нее нажать, укажите формат диска.



» *Déjà Dup* — мощная утилита резервного копирования и восстановления данных из копии, построенная на ветеране Linux, *rsync*. С ее помощью легко делать локальные и удаленные копии нескольких каталогов, и она очень упрощает этот процесс.

Когда будете готовы, нажмите Save [Сохранить], чтобы сохранить изменения настроек и смонтировать диск. Теперь диск будет доступен для подключения по сети. Если у вас Ubuntu или Linux Mint, для подключения к серверу можно воспользоваться файловым менеджером. Также можно воспользоваться и *Gigolo*. (Название, правда, не слишком подходящее. Его девиз — «Монтирует всё, что скажете». Супер.) Это графический клиент для виртуальной файловой системы GIO/GVFs, которая монтирует удаленные файловые системы и открывает их в вашем файловом менеджере. Он есть в репозиториях вашего дистрибутива.

Резервное копирование с Déjà Dup

В последней части этого проекта мы воспользуемся нашим сервером на Pi в качестве устройства для удаленного резервного копирования с превосходной открытой программой под названием *Déjà Dup* (<https://launchpad.net/dejadup>). Фактически это графический клиент для гибкой, быстрой и часто используемой в скриптах *rsync*, которая стала стандартной утилитой Linux с момента своего появления в 1996 г. Она сильно упрощает процесс резервного копирования. Для начала нужно установить *Déjà Dup* на компьютер. На Pi ничего устанавливать не надо.

На компьютере откройте терминал и наберите:

```
sudo apt-get install deja-dup
```

После установки откройте *Déjà Dup*, и вы увидите окно Overview [Обзор]. Вам нужно указать, какие каталоги войдут в резервную копию, и для этого воспользуйтесь опцией Folders To Save [Каталоги для сохранения]. Сделав это, перейдите в меню Storage Location [Местоположение копии] и введите информацию о своем сервере Pi, включая каталог для хранения копии. Мы воспользовались каталогом /media/drive/Documents, который существовал на диске. Сделав это, вернитесь в окно Overview и выберите Back Up Now [Резервное копирование].

Из-за особенностей работы *Déjà Dup* первое резервное копирование займет больше времени, чем последующие. Первая копия — это полная копия выбранных каталогов, а последующие — лишь частичные, которые отражают изменения в файлах. Для восстановления данных из резервной копии, откройте меню Overview, нажмите Restore... [Восстановить...] и следуйте указаниям мастера.

Вот и всё! У нас получился сервер, который требует совсем немного энергии, но служит для размещения всех наших файлов и для удаленного резервного копирования нашей все более разрастающейся цифровой жизни — все благодаря Raspberry Pi и некоторым навыкам Linux.

Чтобы подробнее узнать о *Webmin*, зайдите на www.webmin.com и ознакомьтесь с официальной документацией и решением вопросов в вики. **LXF**

Скорая помощь

К Raspberry Pi можно подключиться удаленно через VNC. Чтобы установить сервер VNC на Pi, подключитесь к нему по SSH, наберите `sudo apt-get install tightvncserver` и затем `sudo tightvncserver` для запуска сервера. Установите программу для просмотра VNC и откройте свой IP-адрес, добавив в конце: 1 для указания порта.

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 Отсутствие настроек беспроводной сети
- 2 Права доступа к каталогу
- 3 Установку openSUSE
- 4 Пакеты для разработчиков
- 5 Монтирование сетевых ресурсов
- 6 Запись действий, выполненных на рабочем столе

1 Без Wi-Fi

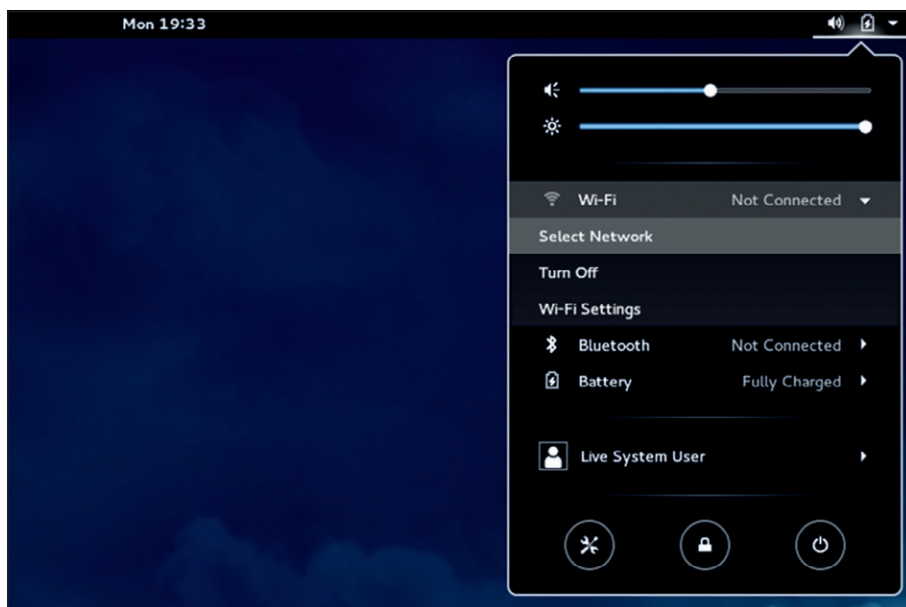
Раньше я мог загрузиться с DVD *Linux Format* для оценки операционной системы, и даже при запуске с DVD Wi-Fi всегда был доступен. Я мог зайти в свой беспроводной роутер и оценить систему. А вот теперь у меня появились проблемы с загруженным образом Mint 17 и с предыдущим DVD Ubuntu 14.10. Ни в одном из них нет выбора беспроводной сети во время установки или после установки; но немного поковырявшись, мне удалось запустить Wi-Fi. В последний раз это случилось с DVD LXF195: Fedora привлекла меня как возможный вариант для двойной загрузки, но от Wi-Fi не было и следа, а с Fedora я не знаком. Я смог подключиться только по проводному Ethernet, но так как я хочу установить дистрибутив на ноутбук с хорошими спецификациями, проводное подключение мне не подходит.

Я пользовался вашими DVD в течение многих лет, и у меня с ними никогда не случалось проблем.

Дэйв Ньюман [Dave Newman]

Если у Вас не самое новое «железо», беспроводная карта должна поддерживаться текущим ядром Linux. Однако это не означает, что среда Live DVD автоматически подключит ее. В отличие от проводных подключений, в которых понятно, к какой сети Вы хотите подключить свой компьютер, беспроводных сетей бывает несколько, и у Live DVD могут возникнуть затруднения с выбором. Беспроводная сеть также нередко защищена паролем, без указания которого Вы не сможете к ней подключиться.

Поэтому первым делом нужно найти иконку, по нажатию на которую откроется окно управления сетью, в котором Вы сможете создать беспроводное подключение. Для рабочего стола Gnome 3 в Fedora 21 в меню нет отдельной иконки, за исключением тех случаев, когда сетевое соединение уже установлено (я лишь опишу положение дел, а не пытаюсь его оправдать).



Настройки сети в Gnome 3 прячутся в общих иконках в верхней части экрана.

Щелкните по ряду иконок в правом верхнем углу экрана, и в открывшемся меню будет пункт для настроек беспроводной сети.

Во всех этих случаях предполагается, что Ваша беспроводная сетевая карта определена и настроена правильно. Может случиться так, что для карты требуется установить файл прошивки, который не установлен по умолчанию.

Файлы прошивки — это файлы с двоичным кодом, необходимые драйверу карты. Часто они «копируются» из драйверов Windows для данной сетевой карты, что в некоторых случаях затрудняет их распространение. Поэтому они не всегда входят в состав дистрибутивов по умолчанию. Если необходимый файл прошивки отсутствует, беспроводной интерфейс вообще не появится. Откройте терминал и внимательно осмотрите вывод команды

```
sudo ifconfig -a
```

Если беспроводного интерфейса нет (его название обычно начинается с "wl", но для некоторых карт Atheros вначале иногда бывает "ath0"), то проверьте сообщения ядра с помощью *dmesg*, например:

```
sudo dmesg | grep wl
```

При наличии сообщения о неудачной попытке загрузки прошивки Вам поможет Ваша любимая поисковая система, хотя для поиска и загрузки прошивки потребуется проводное интернет-соединение. Для многих карт достаточно установить пакет *linux-firmware*.

2 Разблокируем каталог

Мне нужно добавить подкаталог библиотеки (в */usr*) для размещения в нем свеже-созданного символического файла. Программа, которой нужен этот подкаталог — *gschem*, часть комплекта *gEDA*. В конце концов я смог разобраться, как создать новый символический файл, но не могу сохранить его, потому что каталог, в котором должен размещаться этот файл, доступен только для чтения, и изменить это может только root. Я могу переключиться на root в терминале, но не знаю, что делать дальше.

Гарольд Смит [Harold Smith]

Этот каталог доступен вовсе не только для чтения, однако системные каталоги, к которым относится всё в */usr*, должны быть доступны для записи только пользователю root, поэтому для обычных пользователей запись в них обычно заблокирована. Первый вопрос, который нужно задать — действительно ли Вам необходимо сохранять этот файл в */usr*. Хранить в */usr* любые пользовательские и конфигурационные файлы, за исключением файлов программ по умолчанию, довольно необычно. Существует традиционная иерархия для поиска такой информации — сначала файлы по умолчанию в */usr*, затем системные настройки в */etc* и, наконец, пользовательские настройки в домашнем каталоге, и каждое местоположение имеет приоритет над предыдущим. Загляните в документацию

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`.

своей программы и посмотрите, может ли она сохранять свои данные где-то в `$HOME` или в других каталогах, определенных в самой программе.

Если Вам ничего, кроме `/usr`, не подходит, и файлы придется поместить туда, делать это нужно от имени root. Начните с создания файла или каталога в своем домашнем каталоге, затем создайте их содержимое. После этого можно переместить их в `/usr` от имени root командой `mv`, воспользовавшись `sudo`:

```
sudo mv symbols /usr/gscheme
```

В этом примере мы создали в текущем каталоге каталог `symbols`, содержащий необходимые

файлы. Здесь мы этот каталог переносим в `/usr/gscheme`, а Вы подставите путь для своей системы. Если каталог `/usr/gscheme/symbols` уже существует и Вы хотите добавить туда содержимое своего каталога `symbols`, можно скопировать только содержимое каталога, командой

```
sudo mv symbols/* /usr/gscheme/symbols
```

Если Вы хотите сохранить исходные копии файлов — чтобы позднее отредактировать их или добавить к ним другие файлы — воспользуйтесь `cp -a` вместо `mv`: `cp` копирует, а `-a` означает, что нужно скопировать каталоги и их содержимое, сохранив атрибуты файлов.

```
sudo cp -a symbols /usr/gscheme
```

3 Неудача с openSUSE 13.2

В Я загрузил и записал DVD с openSUSE 13.2, но установка не завершилась. Она остановилась на этапе "Installing Boot Manager/saving bootloader configuration [Установка менеджера загрузки/сохранение конфигурации загрузчика]". На этом этапе все файлы были установлены, и после перезагрузки в меню `Grub` появился пункт для openSUSE 13.2.

Обычно после установки новой системы этот пункт появлялся в начале меню, но на сей раз он оказался в конце, наверное, потому, что файлы загрузчика установились не полностью.

Я попытался загрузить установленную систему, но получил следующее сообщение об ошибках:

```
error: no such device [Нет такого устройства]
```

```
error: unknown file system [Неизвестная файловая система]
```

```
error: you must load kernel first [Сперва надо загрузить ядро]
```

Обдумав это, я понял, что большая часть программы, похоже, установилась, и не хватает только правильной записи в меню `Grub`. Поэтому следующим шагом я попытался воспользоваться `Yast` в 12.3, чтобы смонтировать раздел `'var'` из 13.2, и в нем я нашел различные лог-файлы.

Следующие три файла, похоже, относятся к этой установке:

```
/log/zypp/history
```

```
/log/faillog
```

```
/log/lastlog
```

Я приложил первый и последний файл, но скопировать «журнал ошибок [faillog]» не удалось, так как система не смогла его прочесть.

Кристофер Хэггинс [Christopher Huggins]

Вы не можете прочесть этот файл, потому что он доступен для чтения только пользователю `root`. Это двоичный файл, и получить из него информацию непросто. Но из журнала истории понятно, что нужно сделать, чтобы все заработало. Он показывает, что файлы дистрибутива были установлены, и осталось только установить загрузчик. К счастью, в openSUSE для создания файлов `initramfs` теперь используется `dracut`, и `dracut` участливо выводит правильные параметры загрузчика, что и записывается в файле журнала. Всё, что нам нужно теперь — раздел `s /boot`, и Вы уже знаете, что если смонтировали его, то доступ к лог-файлам получите в `/var`. Название раздела нужно перевести с нотации `dev` в Linux в формат `Grub`, который выглядит как `(hdX,Y)`, где X — номер диска (нумерация начинается с нуля), а Y — номер раздела на диске (нумеруются с единицы). Поэтому `/dev/sda3` становится `(hd0,3)`.

Откройте меню `Grub` и измените там неполную запись, которая у Вас уже имеется и должна выглядеть так:

```
menuentry "openSUSE 13.2" {
  set root=(hdX,Y)
  linux /boot/vmlinuz-
  3.16.6-2-desktoppresume=UUID=c0b0bed9-f9cc-
  4754-ad77-425bac1567fa root=UUID=66316bff-
  3fe2-419b-91c3-3c0acb6b9f2b rootflags=rw,relatim
  e,data=ordered rootfstype=ext4 splash=silent
  initrd /boot/initrd-3.16.6-2-desktop
}
```



Коротко про...

/proc

В ы когда-нибудь просматривали свою файловую систему, пытаетесь понять, что занимает так много места на диске, который вроде никогда не заполнялся?

Если да, возможно, Вы видели каталог `proc` в корне файловой системы и невольно задавались мыслью: а нельзя ли обойтись без части из этих тысяч файлов, занимающих более гигабайта места, многие из которых выглядят пустыми?

Ответ — нет, удалить их нельзя. Хорошая новость в том, что места на диске они не занимают, потому что `/proc` — виртуальная

файловая система. Это не набор настоящих файлов на жестком диске, а отражение состояния системы с точки зрения ядра, представляемое в виде файловой системы, чтобы с ним было проще работать.

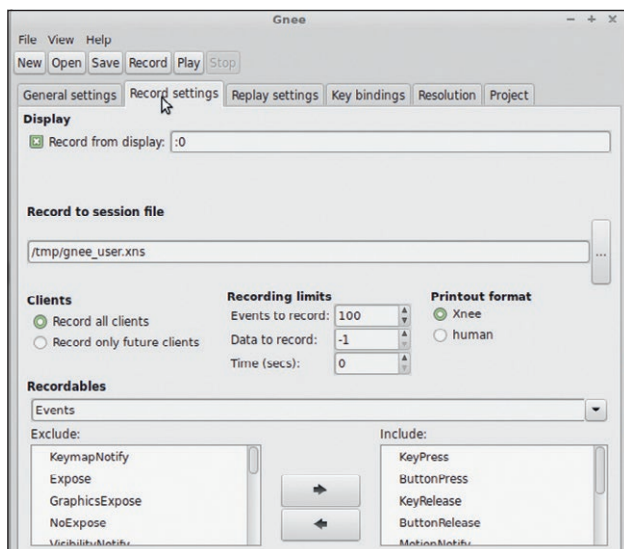
В каталоге `/proc` огромное количество «файлов»: на моем компьютере их в данный момент 7109. Каждый файл содержит частичку информации. Например, чтобы узнать больше о своем дистрибутиве и ядре, выполните команду `cat /proc/version`, а чтобы получить подробную информацию о процессоре — `cat /proc/cpuinfo`. Такие каталоги, как `ide`, `scsi`, `bus/usb`, `bus/pci` и `bluetooth` содержат информацию о подключенных устройствах. Вы также можете обнаружить множество каталогов с номерами. Каждый из них соответствует выполняющемуся

процессу, а его именем является идентификатор процесса. В каждом из этих каталогов вы найдете кучу информации о процессе — его использование памяти, запустившую его команду и многое другое.

Большая часть содержимого `/proc` доступна только на чтение, но есть несколько файлов, которые можно записывать. Например, `/proc/sys/vm/swappiness` определяет, насколько активно система будет выбрасывать программы в раздел подкачки: большее значение означает более активный сброс. Значение можно изменить командой `echo`:

```
echo 30 >/proc/sys/vm/swappiness
```

Подробную информацию по `/proc` можно найти в файле `/usr/src/linux/Documentation/filesystems/proc.txt`.



С помощью *Gnee* и его анапарника в командной строке *snee* можно записать действия на рабочем столе в файл для последующего воспроизведения.

Часть от `resume` до `ext4` — командная строка ядра, которая находится ближе к концу лог-файла истории, и Вы можете просто скопировать ее оттуда, вместо того чтобы набирать длинные UUID.

Хотя после этого система начнет нормально загружаться, тот факт, что при установке вообще произошла ошибка, беспокоит. Может быть, файловая система, содержащая исходный раздел `/boot`, переполнена или повреждена? Это можно проверить командами

```
df /boot
sudo fsck /dev/sdaN
```

То, что несколько попыток установки оборвались на одном и том же месте при попытке записи в меню *Grub*, указывает на некую проблему, из-за которой запись не удалась. openSUSE сначала должна установиться сама и затем установить загрузчик безо всяких проблем. Повторные ошибки на одной и том же этапе могут говорить об аппаратной проблеме или проблеме с файловой системой. Возможно, она просто переполнена. Если `/var` находится в той же файловой системе, проверьте наличие больших файлов в `/var/log/`. Вышедший из-под контроля процесс иногда способен заполнять системный лог-файл с огромной скоростью.

4 Неясность с dev-пакетами

При установке программы через *Synaptic* у меня свободно всего 32 Гб на диске. Можно ли не устанавливать файлы пакетов для разработчиков (dev-)? Буду ли я неправ, предположив, что моя система самостоятельно сможет отлаживать себя, создавать новые исправления и т.д.? Может, она просто создаст лог-файлы для загрузки в Ubuntu при каждом сбое? Я не знаю ответа и не могу найти его в Google.

Alex01UK, С форумов

Короткий ответ чаще всего — «доверьтесь своего менеджеру пакетов». Установите ту программу, которая Вам нужна,

и предоставьте менеджеру пакетов заботу по установке библиотек и других зависимостей, необходимых этой программе. Если Вы приметесь устанавливать библиотеки отдельно, то запутаете менеджера пакетов, который по Вашей просьбе следит за тем, что устанавливается и что устанавливается как зависимость. Например, он может удалить пакеты, от которых больше не зависит ни одна программа.

Большинство библиотек поставляются в двух пакетах, например, *libfoo* и *libfoo-devel*. Первый пакет содержит собственно библиотеку, а второй (пакет для разработчиков) содержит заголовочные файлы исходного кода

для этой библиотеки. Обычно они Вам не нужны. Единственный случай, когда они могут потребоваться — если Вы хотите скомпилировать из исходников какую-то программу, которой нужна эта библиотека; в этом случае ей понадобятся заголовочные файлы.

Вы увидите это, если на этапе настройки компилируемого пакета появится сообщение о том, что *libfoo* не найдена, даже несмотря на то, что она установлена. Это практически единственный случай, когда придется установить пакет для разработчика вручную. Это не связано с общим обслуживанием системы. Обновления загружаются менеджером пакетов как обычно — пакеты разработчика нужны людям, разрабатывающим эти обновления.

С переходом на SSD мы снова возвращаемся к тому, что у нас «всего» 32 Гб. (Кто-нибудь еще помнит времена, когда диски в 1 Гб считались огромными? Теперь диском в тысячу раз больше никого не удивит!) Так или иначе, файлы для разработчиков содержат только заголовочные файлы для библиотек — это небольшие текстовые файлы, и они займут совсем немного места на Вашем жестком диске.

Если Вам на самом деле нужно установить какие-то пакеты, то достаточно будет удалить одну фотографию или песню, и у Вас появится более чем достаточно свободного места для данной цели.

5 Немонтируемые ресурсы

У меня Fedora 21, в которой используется *systemd*. Должен сказать, меня впечатлило увеличение скорости загрузки, но есть проблемы с сетевыми ресурсами — при загрузке ресурсы с моего NAS не монтируются.

Я отключил экран заставки при загрузке, чтобы увидеть загрузочные сообщения, однако *systemd* загружается так быстро, что осознать происходящее невозможно. Я думаю, что он пытается смонтировать мои сетевые ресурсы

перед настройкой сети. Открыв после загрузки терминал, я могу монтировать эти ресурсы вручную безо всяких проблем, поэтому проблема, наверное, в том, что *systemd* делает нечто неправильное, препятствуя монтированию ресурсов.

Тревор Бойс (Trevor Boyce)

Вы правы в том, что это действительно проблема *systemd*, но решить ее просто. В *systemd* почти все, включая монтирование файловых систем, определяется файлами модулей. В самом начале загрузки *systemd-fstab-generator* создает файлы `.mount` для каждой из записей в `/etc/fstab`. Как и у всех модулей *systemd*, у этих модулей могут быть зависимости, которые определяют их запуск до или после других модулей. Локальные ресурсы монтируются на ранней стадии, а удаленные не монтируются до тех пор, пока не будет загружен сетевой стек. Однако между инициализацией сети и ее фактической доступностью может пройти несколько секунд, и удаленное монтирование выполняется слишком рано и завершается неудачно.

Посмотреть, завершилось ли оно неудачно, можно, выполнив следующую команду после загрузки:

```
journalctl -b -p err
```

Удаленное монтирование зависит от модуля под названием `network.target`. Этот модуль означает, что будет запущена только программа для управления сетью, а не то, что сеть будет фактически куда-то подключена. Также есть модуль `network-online.target`, для работы которого необходимо по крайней мере один IP-адрес. Как этим воспользоваться, зависит от способа, которым осуществляется управление сетью. В большинстве дистрибутивов используется *Network Manager* [сетевой менеджер], и у него есть специальный сервис, запускающий `networkonline.target`. Так как для работы `network.target` необходима программа управления, в данном случае, *Network Manager*, это означает, что система не будет пытаться смонтировать ресурсы до тех пор, пока сеть не заработает фактически. Чтобы включить его, выполните команду

```
sudo systemctl enable NetworkManagerwait-online.service
```

Если Вы пользуетесь собственной утилитой *systemd* для управления сетью, *networkd*, то у нее имеется эквивалент, который можно включить командой

```
sudo systemctl enable systemd-networkdwait-online.service
```

Пользуйтесь этим осторожно, если у Вас ноутбук, который подключается к различным сетям — попытки монтирования Ваших сетевых ресурсов при подключении к другой сети вызовут задержки загрузки, особенно если Ваш сетевой менеджер не может автоматически подключиться к сети (или если нет доступной сети). В таком случае, возможно, будет проще добавить `noauto` к параметрам монтирования каждого ресурса в `/etc/fstab`, чтобы отключить их автоматическое монтирование, и затем смонтировать их вручную, когда Вы подключитесь к подходящей сети.

6 Повтор рабочего стола

В Мне часто нужно продемонстрировать то, как я пользуюсь рабочим столом в Linux, и я хотел бы это автоматизировать.

Я знаю, что возможно записать эти действия как видеоролик, но хотел бы выяснить, можно ли записать действия клавиатуры и мыши как макрос и воспроизвести позже. Тогда я мог бы скопировать макрос на другие компьютеры и частично продемонстрировать работу с рабочим столом, а затем позволить обучающимся продолжить работу, что, разумеется, было бы невозможно с видеозаписью.

Есть ли какая-нибудь программа, которая способна мне помочь?

Карл Тернбалл [Carl Turnbull]

Программа, которая Вам нужна — *Xnee*, и ее можно загрузить с <https://xnee.wordpress.com>. В состав *Xnee* входят три программы (и ни одна из них не называется “xnee” — это название библиотеки): *snee*, *gnnee* и *plnee*. Это утилита командной строки, ее графический эквивалент и апплет для панели Gnome соответственно. Для двух последних необходимо установить библиотеки рабочего стола Gnome, а версия для командной строки работает на любом рабочем столе с X.

Вот простой пример:

```
snee --record -time 1 --mouse --keyboard
--seconds-to-record 30 --out-file macro1
```

Параметры не требуют пояснений, возможно, за исключением **-time**: это время ожидания перед началом записи. Это важно при запуске *snee* из терминала, иначе первым действием, которое она запишет, станет отпущение клавиши Enter. Другие параметры переводят *snee* в режим записи и сообщают утилите о том, что нужно записывать действия с клавиатуры и мыши в течение 30 секунд и сохранить результат в файл. Воспроизвести этот файл Вы сможете командой

```
snee --replay --file macro1
```

Графическая программа работает похожим образом, но *snee* можно запустить с помощью горячих клавиш, чтобы записывать действия, не открывая окна программы. Есть масса параметров для тонкой настройки вывода, среди которых стоит взглянуть на **--first-last**, он сглаживает перемещения мыши. Если взглянуть на файлы макросов, которые создает *Xnee*, Вы увидите, что это текстовые файлы, и если что-то в них будет не так, их можно будет отредактировать.

Единственный момент, за которым стоит следить — при записи и воспроизведении событий мыши предполагается, что окна каждый раз будут открываться в одном и том же положении, в противном случае макрос будет щелкать в неправильное место. Чтобы это сделать, нужно открыть окна и развернуть их на весь экран. Также в начале каждой записи следует перемещать мышь в левый верхний угол экрана, чтобы она всегда имела одну и ту же стартовую точку. **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

Брандмауэры

» **Что такое брандмауэр и почему я должен беспокоиться о том, что мой компьютер загорится?**

Брандмауэр — сочетание устройства и программы, которые предотвращают неавторизованное использование защищаемой ими сети. Самое частое его применение — защита компьютера или сети от нападений менее приличных обитателей Интернета. К сожалению, это не защитит компьютер от возгорания.

» **Устройства и программы? Брандмауэр — это программа или отдельный ящик?**

Это может быть и то, и другое. Отдельный ящик обеспечивает лучшую защиту, так как останавливает всех злоумышленников еще до того, как они достигнут вашего компьютера (компьютеров). Во многих широкополосных роутерах есть встроенный брандмауэр, поэтому плохих

парней можно тормознуть у вашего модема. Другой вариант — старый компьютер с дистрибутивом брандмауэра, таким как *IPSec*. Для работы брандмауэра не нужен мощный компьютер: достаточно старого 486.

» **У меня один компьютер, можно ли запускать брандмауэр на нем?**

Да, можно, и довольно эффективно. Хотя программный брандмауэр не так защищает систему, как аппаратный, он все равно отлично удерживает злоумышленников. Конечно, истинные параноики запускают и аппаратный, и программный брандмауэры.

» **А какие бывают варианты брандмауэров?**

Сами задачи брандмауэра в Linux выполняются *Iptables*, частью ядра. Для их настройки можно написать набор правил, определяющих,

что разрешено и что нет. Написание правил *Iptables* — довольно хитрая задача, и в результате можно получить брандмауэр, который не будет выполнять того, что вы ожидали, или не будет выполнять вообще ничего. Существует несколько программ брандмауэров. Все они, от *Shorewall* (www.shorewall.net) на основе скриптов до графических программ, таких как *Guarddog* (www.simonzone.com/software/guarddog) и *Firewall Builder* (www.fwbuilder.org) упрощают создание правил. Все они генерируют скрипты *Iptables*. Также можно воспользоваться одной из графических программ для генерации скриптов брандмауэра, которые будут применяться на сервере без X.

» **Но разве Linux не безопасен?**

Зачем мне тогда брандмауэр?

Безопасен, но только если вы возьмете на себя труд сделать и поддерживать его таким. Хороший дверной замок не остановит вора, если вы забыли закрыть окно. То же

справедливо и для безопасности в Интернете. В Linux есть утилита, которая надежно защищает сеть, но только в том случае, если вы ими пользуетесь. Пользовательские права доступа затрудняют действия злоумышленника, но лучше удерживать его вне вашей системы.

» **У меня ведь нечего красть, не правда ли?**

Больше, чем вы можете представить. Например, по кэшу браузера можно понять, какие финансовые сайты вы посещаете. В нем не будет паролей, но эта информация даст вору ценные сведения. Ваши электронные письма содержат персональную информацию. Затем есть задача воспользоваться вашим интернет-соединением. Спамер может воспользоваться им для рассылки спама, что может привести к закрытию вашей учетной записи провайдера. Ни одно из этих действий не требует доступа администратора root, достаточно прав доступа обычного пользователя.



LXF HotPicks



Александр Толстой, он же DJ Толстой, постоянно вбрасывает в вечеринку все новые релизы программ, подбирая новую сессию хитов чартов открытого кода.

Shotwell » Linphone » Fiziko » Audacity » Brackets » Netsurf » Geary
» Pasang Emas » FreeCol » VirtualBox » FreeMind

Организатор фото

Shotwell

Версия: 0.22 Сайт: <http://bit.ly/GnomeShotwell>

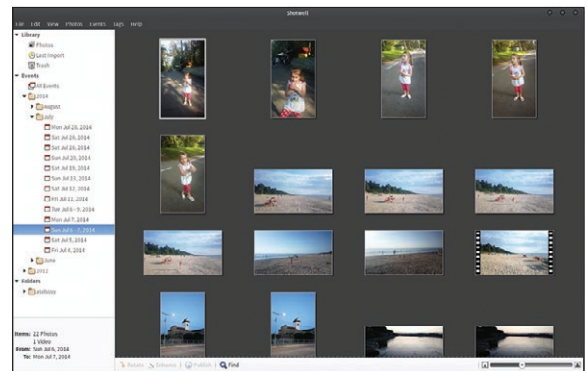
Shotwell существует с 2009 г., когда он заменил *F-Spot* как приложение по умолчанию для управления фотографиями в Gnome. Это легковесное и опрятное приложение *GTK*, создававшееся с упором на простоту, также предлагает простейшие функции обработки фотографий.

Например, *Shotwell* умеет импортировать фото и видео напрямую с цифровой камеры, используя режим массового хранения USB или режим MTP. Импортируемые файлы сохраняются в директории **Pictures**, и *Shotwell* автоматически группирует фото и видео по дате и отображает их с большими эскизами, как и полагается

менеджеру файлов фотографий. Левая панель используется для навигации между альбомами, созданными по дате, а основное окно показывает содержимое текущего альбома.

Когда менеджер файлов *Nautilus* исчез из последних релизов, *Shotwell* взял на себя задачу просмотра фото. Это делается благодаря его привычному древовидному отображению и быстрому созданию

«Автоматически группирует фото и видео по дате и отображает.»



► Быстрый и достаточно мощный инструмент для работы с фото, в котором достаточно функций, чтобы для большинства пользователей заменить *Digikam*.

эскизов, которые поддерживают видео и даже фото RAW. *Shotwell*, конечно же, способен на большее: он может организовать ваши файлы по времени, тэгам (ключевым словам) и папкам, и осуществить поиск и удаление дубликатов. Он также позволяет выполнять основные задачи по расширению и редактированию (например, обрезать фото, повернуть, настроить, поработать с цветом, выровнять и т.д.), просматривать слайд-шоу и делиться фотографиями на основных web-сервисах, включая Facebook и Flickr.

В версии 0.22 разработчики *Shotwell* рядом способов улучшили поддержку метаданных: теперь загрузка ваших фото на Facebook поддерживает тэги и подписи, и приложение также может импортировать время выдержки и название фото, при наличии таковых.

Более того, *Shotwell 0.22* предлагает массу других улучшений и отладок, например, нормальную поддержку опций текста is-set и not-set при сохранении результатов поиска, возможность сортировать фото по имени файла и мелкие косметические улучшения GUI.

Для Ubuntu и его большого семейства Yorba Foundation предлагает PPA (ppa.yorba.org), который поддерживает все современные релизы. В других дистрибутивах тоже появился *Shotwell* вместе с новым набором приложений Gnome, так что проверьте свой менеджер пакетов на предмет обновлений.

Исследуем интерфейс Shotwell

Панель навигации

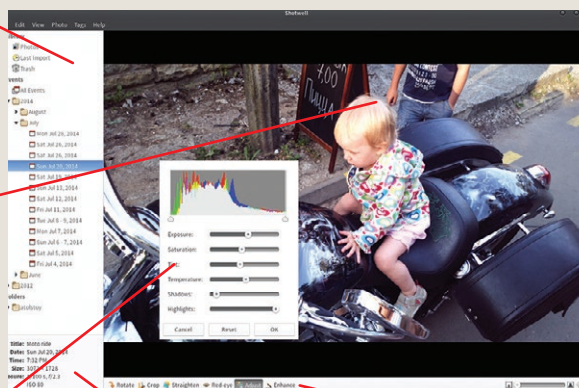
Просматривайте свою коллекцию фото либо как Events с тэгами, либо в виде простого дерева Folders.

Наслаждайтесь общей картиной

Shotwell может показать выбранное вами изображение в оконном или в полноэкранном режиме. (Нажатие F11 сменит режим).

Настройка цвета

Имеется режим автоматической оптимизации цвета, однако предусмотрена и ручная настройка, с «живой» гистограммой.



Данные EXIF

В этом углу хранятся переменные EXIF для Date [Даты], Time [Времени], Size [Размера] и Exposure [Выдержка], а также заполняемое пользователем поле Title [Названии].

Простейшее редактирование

Поворачивайте, кадрируйте и выравнивайте свое изображение и оптимизируйте его вручную или в автоматическом режиме.

SIP-клиент

Linphone

Версия: 3.8 Сайт: www.linphone.org

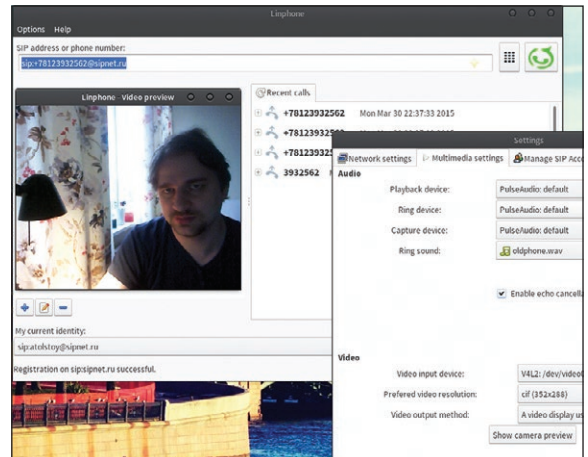
Потребность в высококачественном VoIP-решении для Linux трудно переоценить. У нас есть проприетарный и непредсказуемый Skype, многообещающий, но пока недостаточно зрелый Hello! в Firefox и ряд других хороших решений, таких, как Tox или Jabber, которые работают, но отнюдь не идеальны.

Linphone — еще один коммуникатор VoIP/SIP или «софтфон», как их иногда называют. Это кросс-платформенное решение с открытым кодом, разработанное Belledonne Communications из Франции. Когда заходишь на их сайт, появляется такое «корпоративное» ощущение, из-за которого трудно поверить, что это — совершенно свободное ПО; но оно доступно для всех основных настольных и мобильных ОС, включая Android и Linux.

Приложение применимо для аудио- и видеозвонков и обмена текстовыми сообщениями, и в дополнение к настольному клиенту имеется также сетевой сервис Linphone и мощный SDK для SIP VoIP видео

под названием Liblinphone, для разработки сторонних решений.

Обычным людям Linphone эффективно заменит Skype, особенно для бесплатных звонков между компьютерами или мобильными телефонами. Linphone включен в репозитории Ubuntu и Debian по умолчанию (`sudo apt-get install linphone`), и <http://pkgs.org> указывает, что многие другие дистрибутивы Linux тоже идут с Linphone. Чтобы его использовать, надо обзавестись учетной записью SIP, которая может быть предоставлена любым провайдером SIP. Если у вас ее нет, Belledonne предлагает собственный бесплатный сервис SIP (@linphone.org), который вы можете использовать, когда запустите Linphone в первый раз (появится мастер). Войдя



Linphone предлагает высококачественные видеозвонки и будет работать на всех основных настольных и мобильных ОС.

в программу, вы получаете высококачественные видеозвонки (поддерживаются многие современные кодеки), управление звонками, адресную книгу, устранение эхо, автоматическую настройку скорости звукопередачи и множество других профессиональных функций. В новой версии 3.8 появился полноэкранный режим, поддержка кодека VP8, инструмент настройки аудио и видео и масса мелких улучшений.

«Применимо для аудио- и видеозвонков и текстовых сообщений.»

Пакет MetaPost

Fiziko

Версия: 0.1 Сайт: <http://bit.ly/Fiziko>

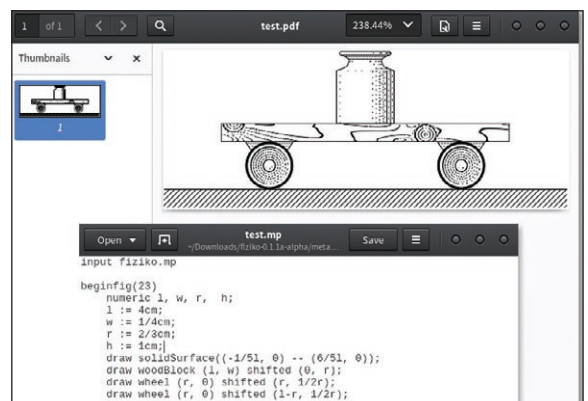
Многие предпочитают программы *Tex* и *LaTeX* более причудливым текстовым редакторам WYSIWYG, особенно когда речь идет о научном исследовании. *Tex*, как известно, отлично подходит для работы с математическими или химическими формулами, а также предлагает очень точный контроль за форматированием текста, программных и текстовых ссылок и т.д. Долгие годы *Tex* сопровождал язык программирования MetaPost и интерпретатор языка Metafont от Дональда Кнута [Donald Knuth]. MetaPost отлично справляется с созданием диаграмм в языке программирования PostScript из геометрического/алгебраического описания, используя элегантный декларативный синтаксис для работы с линиями, кривыми, точками и геометрическими преобразованиями.

Fiziko — это пакет MetaPost, изначально разработанный для облегчения подготовки иллюстраций к учебникам

физики, и включает набор «низкоуровневых», да и «высокоуровневых» макросов для создания красивых черно-белых иллюстраций.

Пакет основан на той идее, что разрешения печати часто бывает мало для сокрытия уродливых растровых переходов тонов, и даже при достаточном разрешении исходные цифровые оттенки градиента выглядят ненатурально и скучно. И вот, вместо использования обычных полутонов, Fiziko имитирует гравюру на дереве или рисунок тушью тонкой штриховкой, которая придает иллюстрациям старинный вид. Меняя толщину штриха, можно настроить иллюстрации на быструю печать на оборудовании с ограничениями

«Имитирует гравюру или рисунок тушью тонкой штриховкой.»



Fiziko стилизует рисунки под старину, и при печати они выглядят очень аккуратно и точно.

или без таковых, или с минимальными изменениями в самих иллюстрациях.

Fiziko содержит несколько макросов, дающих доступ к некоторым простейшим объектам, таким, как кривые, тележки, блоки, глаза, блоки, колеса, и другие. Следуя легко читаемому коду MetaPost, вы создаете сложные сочетания из набора простых объектов (например, тележка на рисунке). Закончив со своим `_file.mp`, конвертируйте его в PostScript (`mpost your_file.mp`) и затем в PDF (`ps2pdf your_file.1`), где 1 — номер фигуры из параметра 'beginfig'.

Аудиоредактор и программа для записи

Audacity

Версия: 2.1 Сайт: <http://audacity.sourceforge.net>

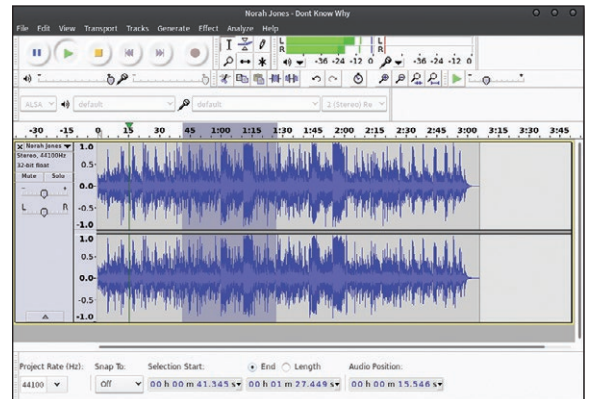
Audacity — любимая тяжеловесная аудиопрограмма для Linux, хотя за последние годы она нечасто упоминалась в новостях открытого кода. Причина в том, что последняя версия (2.1) была в разработке более трех лет, и вот наконец-то вышла в конце марта.

Audacity — многодорожечный аудиоредактор и программа для записи с открытым исходным кодом, основанная на GTK и wxWidgets. Она может записывать поток с микрофона или даже то, что воспроизводится на компьютере. Все операции по вводу/выводу производятся или через классическую ALSA, или через современную систему PulseAudio. Вы также сможете импортировать аудиоданные, просто открыв свои WAV, MP3, FLAC или любой другой поддерживаемый формат (их десятки). Где бы вы ни взяли аудио, оно воспроизводится в виде графика на временной шкале, которую вы редактируете с помощью разных инструментов. Верхняя часть окна Audacity размещает

строку воспроизведения, управление вводом/выводом и набор простейших инструментов редактирования. С их помощью можно увеличивать и уменьшать график, редактировать его пики и даже вручную «рисовать» звук.

Audacity отлично подходит для микширования треков; например, ее можно использовать для создания домашних подкастов, мастеринга ваших любительских записей и ремиксов. Сильной стороной редактора является щедрый набор встроенных эффектов в меню Effects. Можно поэкспериментировать с изменением темпа, затуханием, растягиванием, добавлением эха и поиском звука, который хочется слышать. Кроме этого, Audacity — необходимый инструмент для реставрации

«Необходимейший инструмент для реставрации записей.»



➤ Все эти годы интерфейс Audacity был интуитивным и простым в использовании, а сейчас он еще похорошел.

и исправления поврежденных записей: здесь имеются расширенный фильтр для шумоподавления, функция устранения щелчков, нормализатор и ряд другие мелких функций, помогающих вдохнуть новую жизнь в старые записи.

В новой версии добавлен долгожданный Live Preview [Предпросмотр] для фильтров, значительно улучшен фильтр шумоподавления и модернизирован GUI, и вы увидите, что Audacity — часть практически любого основного дистрибутива Linux. Когда вы будете это все читать, версия 2.1 уже должна появиться в репозиториях обновлений.

Текстовый редактор

Brackets

Версия: 1.2 Сайт: <https://github.com/adobe/brackets>

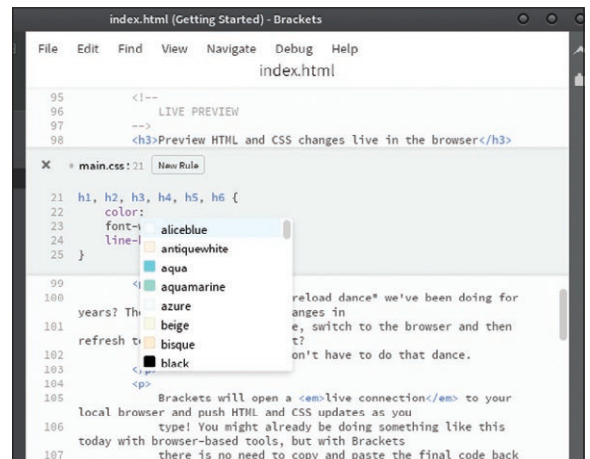
Adobe Systems впервые заговорили о своем текстовом редакторе с открытым исходным кодом (под лицензией MIT) летом 2012 г. (вместо с новостью о прекращении поддержки Flash для Android). Первый стабильный релиз Brackets увидел свет в конце ноября 2014 г., но тогда не вызвал особого интереса в сообществе Linux по причине отсутствия в нем каких бы то ни было прекомпилированных пакетов, кроме как для Ubuntu (ppa:webupd8team/brackets). В последние месяцы это исправили, и теперь Brackets стал намного доступнее в Linux.

Редактор создан для веб-дизайна и разработки, и использует технологии HTML, CSS и JavaScript. При запуске он кажется очень аккуратным и милым, слегка напоминая Adobe Photoshop. На левой панели показан список недавних файлов, размещенный поверх списка рабочей директории, которую вы должны определить вручную. Основная часть — это сам

редактор, который сопровождается панелью наверху.

В Brackets есть несколько крутых функций, которые делают веб-разработку не столь трудоемкой. Например, когда вы вводите тэг <color> в CSS, всплывает автоматическая подсказка с W3C-совместимыми именами <color> и образцом. При щелчке по тэгу в HTML и нажатии Ctrl+E появляется соответствующий CSS код, и вы сможете мгновенно изменить его, не отключаясь от своего файла. Основные подсказки по коду поддерживают большинство известных языков разметки, включая, например, SVG. Справа есть узкая вертикальная панель с двумя дополнительными функциями: «живым»

«Несколько функций делают веб-разработку не столь трудоемкой.»



➤ В Brackets масса функций дизайна, которые позволят избавиться от многих скучных рутинных задач.

предпросмотром и менеджером расширений. Предпросмотр передает ваш код в браузер (пока только Chrome), а менеджер расширений открывает окно, из которого вы можете установить множество добавочных функций и тем от огромного сообщества Brackets.

Новая версия предлагает лучшую поддержку подсказок, поддержку текстового drag-and-drop и улучшения интерфейса, и это удобный инструмент для веб-разработчиков и фанатов кода.

Web-браузер

Netsurf

Версия: 3.3 Сайт: www.netsurf-browser.org

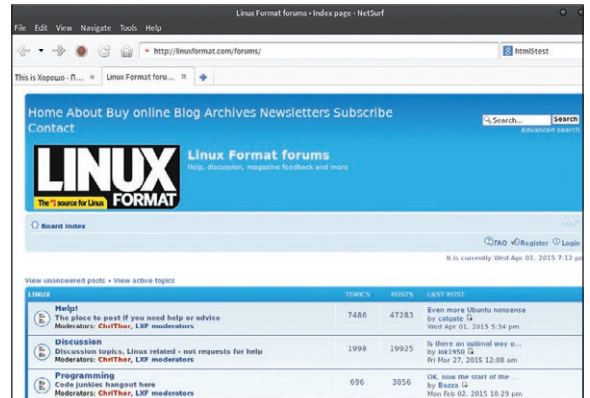
Пока не утихают широко обсуждаемые битвы между *Chrome* и *Firefox*, многие второстепенные браузеры оказываются практически заброшенными, оставаясь в тени битвы гигантов. Но отойдя чуть в сторонку от этого ристалища, мы обнаружим *Netsurf*, веб-браузер с собственным настраиваемым движком рендеринга HTML, который отличается легковесностью, невероятной быстротой и имеет поддержку вкладок, журнал, предварительный просмотр страниц и прочие современные функции.

Netsurf сильно отличается от прочих браузеров, с которыми вы могли повстречаться; он появился в 2002 г. как браузер для RISC OS. В 2004 г. его портировали на *GTK*, и он стал доступен в Linux. С тех пор команда разработки реализовала поддержку разнообразных web-стандартов; в первую очередь — HTML 4/5 и CSS. Сегодня поддержка стандартов вышла на приличный уровень, хотя надо признать, что еще многое предстоит сделать.

По-прежнему здесь не хватает поддержки JavaScript и многих функций HTML5; но в остальном браузер работает очень хорошо. Например, в нем имеется автодополнение URL в адресной строке, поддержка масштабирования страниц, HTTPS, cookie, сохранение страниц с изображениями и многое другое.

Netsurf также весьма скромнен по части места на жестком диске и потребления ОЗУ. Мы обнаружили, что при посещении большинства сайтов потребуется всего лишь 15–25 МБ. Страницы обрабатываются с помощью собственных библиотек проекта *Hubbub*, *LibCSS* и *LibDOM*, причем очень быстро. Самая последняя версия, 3.3, разрабатывалась около 10 месяцев и продолжает реализовывать подход,

«Скромнен по части места на жестком диске и потребления ОЗУ.»



➤ Большинство web-страниц отображаются в *Netsurf* корректно, однако не все.

предложенный разработчиками *Netsurf* в ноябре 2014 г.: реализация поддержки стандартов DOM, JavaScript, прокси (помимо HTTP), тем *GTK* и большего числа свойств CSS.

Чтобы познакомиться с *Netsurf*, надо скомпилировать его из исходника. Процесс этот не особо сложен, и благодаря отличному руководству Quick Start Guide (<http://bit.ly/1IPgxMq>) каждый шаг задокументирован. На данный момент предоступленные бинарники доступны только для Arch Linux (в AUR).

Почтовый клиент

Geary

Версия: 0.10 Сайт: <http://bit.ly/GearyMailClient>

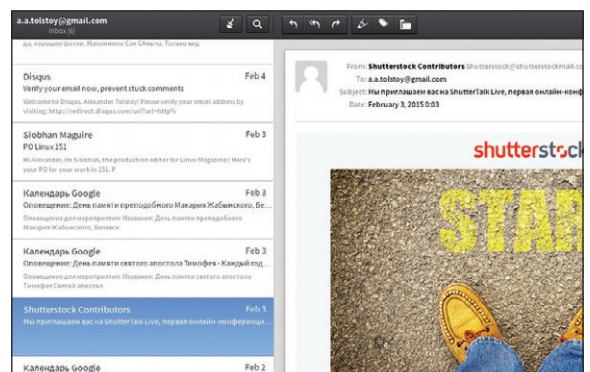
У Orba Foundation в последнее время приналег на продуктивность, и уже во второй раз попадает в Hotpicks этого месяца. У *Geary*, минималистского, стильного и быстрого почтового приложения, вышел новый релиз, версия 0.10. Это клиент IMAP, первоначально разработанный для рабочего стола Gnome в стиле *GTK3*. *Geary* написан на Vala и использует базу данных *SQLite3* для хранения электронной почты и библиотеку *GObjectbase* для доступа к папкам IMAP. По подключении к *Geary* хотя бы одной учетной записи электронной почты автоматически создается и поддерживается текстовый индекс, позволяющий осуществлять очень быстрый поиск по ключевым словам по всему почтовому ящику, где цепочки сообщений отображаются в виде диалогов.

У интерфейса *Geary* обычная трехпанельная раскладка, как у большинства web-приложений и клиентов, подобных

Thunderbird. В левой панели — список ваших учетных записей, и входящие для них размещаются в верхней части столбца. Под входящими находятся расширяемые индексы всех папок вашей учетной записи. Выберите папку, и центральная панель отобразит находящиеся в ней сообщения. Выберите сообщение из списка, и правая панель покажет выделенное сообщение.

Geary 0.10 предлагает полноценный почтовый клиент со своими новыми функциями, среди которых — поддержка отмены предыдущего действия для перемещения, удаления и архивирования почтовых сообщений; улучшенная раскладка верхней панели; новый дизайн составителя сообщений и возможность приписывать

«Предлагает полноценный почтовый клиент с новыми функциями.»



➤ *Geary* — отличное приложение от друзей *GTK3* и трудолюбивых разработчиков проекта *Elementary*.

более одного почтового ящика к одной учетной записи. Помимо всех этих функций, *Geary* также имеет весьма достойный составитель почтовых сообщений WYSIWYG и HTML (на основе *WebKitGTK*), уведомления рабочего стола, быструю настройку учетных записей для Gmail, Yahoo! Mail, Outlook.com и популярных серверов IMAP (*Dovecot*, *Cyrus*, *Zimbra* и т.д.) и полную поддержку SSL и STARTTLS.

Geary особо легко устанавливается на ОС Ubuntu и на ее производных (ppa:yorba/ppa), и доступен в Arch (AUR) и многих других разновидностях Linux. Он отлично вписывается в ваш рабочий стол на *GTK*.

HotGames Развлекательные приложения

Настольная игра

Pasang Emas

Версия: 5.0.0 Сайт: <http://bit.ly/PasangEmas>

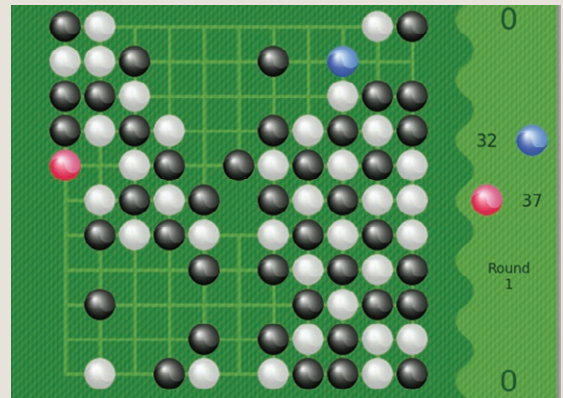
Рasang Emas — традиционная настольная игра из Брунея, где используется доска 10×10 с 60 белыми и 60 черными фишками. Во время игры два игрока ходят, набирая очки захватом фишек и стараясь не застрять.

Однако правила довольно сложны, и прежде чем приступить к игре с ИИ, стоит их выучить. Игра начинается с размещения фишек в некий узор — симметрично вокруг центра поля, где оставляется пустая клетка. Игроки создают для себя проходы, выбирая любой вертикальный ряд из пяти фишек. Когда первый игрок создает ряд, второй игрок может выбрать любой ряд, кроме соответствующего симметричного ряда со своей стороны. Когда у обоих игроков есть проходы, они продвигают любую фишку, выбранную из группы в две колонки с каждой стороны прохода.

Продвинутое фишки перемещаются или перепрыгивают в проход и превращаются в «Кас» — подвижную фишку-дамку, которая в дальнейшей игре будет захватывать больше фишек.

Игроки различают свои фишки Кас по цвету; обычно они красные и голубые. Во время каждого хода игрок перемещает свою Кас по горизонтали или вертикали, но чтобы ход был законным, он должен образовывать линию с нечетным числом фишек того же цвета. Линия может пересекать пустые клетки, но не может прерываться фишками другого цвета. Чтобы победить, игрок

«Игроки ходят по очереди, набирая очки захватом фишек.»



► Нужно предугадывать следующие стадии игры, так же, как в Reversi.

должен набрать больше очков с каждой белой фишкой, которая стоит 2 очка, а черная фишка стоит 1 очко.

Игра заканчивается, когда все фишки захвачены или когда у игрока заканчиваются ходы и он оставляет фишку не захваченной, и тогда выигрывает его противник.

Pasang Emas имеется в пакетах для Fedora и Arch, но в остальных дистрибутивах его придется компилировать из исходника.

Пошаговая стратегия

FreeCol

Версия: 0.11.3 Сайт: <http://bit.ly/FreeColGame>

Как явствует из ее названия, FreeCol — реализация с открытым кодом известной стратегии Colonization, вышедшей в 1994 г. Игра начинается в 1492 г. с целью колонизации Америки. Вы начинаете игру посреди океана при одном корабле с двумя колонизаторами, и бороздите волны, пока не достигнете неизведанной земли.

Так же, как и в Civilization (еще одной знаменитой стратегии от Сиды Майера [Sid Meier], чей клон с открытым кодом именуется Freeciv), вам нужно создать мощную нацию, но на ранней стадии игры вы сможете отправлять корабли обратно в Европу, чтобы продавать произведенные вами товары или привозить колонистов. Заселение нового мира колонистами — очень важный аспект игры, поскольку один игровой ход соответствует одному году, а потом и одному сезону. В результате колонии

растут не так быстро, как в Civilization. Вы можете заплатить колонистам за переезд в Новый Свет или соблазнить их обещанием свободы религии, и тогда они совершенно бесплатно сами взойдут на ваши корабли.

FreeCol имеет ряд улучшений по сравнению с игрой-оригиналом: она поддерживает многопользовательский режим и показывает карты в изометрии. Игра кросс-платформенная и основана на Java, и для Linux она распространяется в виде программы установки JAR. Чтобы установить и запустить ее, убедитесь, что у вас есть OpenJDK, а затем введите `java -jar freecol-0.11.3-installer`.

«FreeCol имеет ряд улучшений по сравнению с оригиналом.»



► Вместо разжигания войн, вы можете подружиться с другими европейцами, которые колонизируют Америку.

jar и следуйте указаниям мастера установки. По умолчанию игра устанавливается в директорию `~/freecol`, где вы найдете скрипт запуска (`./freecol`) и даже значок FreeCol.

Начните новую игру и выберите нацию, или присоединитесь к многопользовательским сетевым матчам, или самостоятельно запустите общественный сервер, используя флажок Public в окне новой игры. Игра работает в оконном режиме и неплохо идет даже на медленных компьютерах.

Программа виртуализации

VirtualBox

Версия: 5.0.0 beta1 Сайт: <http://bit.ly/VirtualBox5B1>

Уже несколько месяцев нарастает беспокойство от того, что разработка знаменитой программы *VirtualBox* приостановилась (или крайне замедлилась), и мы уже давненько не слышали о серьезных инновациях или улучшениях в *VirtualBox*. Собственно, с тех самых пор, когда Oracle приобрели Sun Microsystems в 2010 г.

Хотя *VirtualBox* поддерживается всей корпоративной мощью Oracle, это программа с открытым кодом, что оставляет надежду увидеть улучшения, и создается впечатление, что наступило время перемен: Oracle обнародовала бета-версию *VirtualBox 5.0* и вместе с ней — поддержку паравиртуализации для гостей Windows/Linux для улучшения производительности.

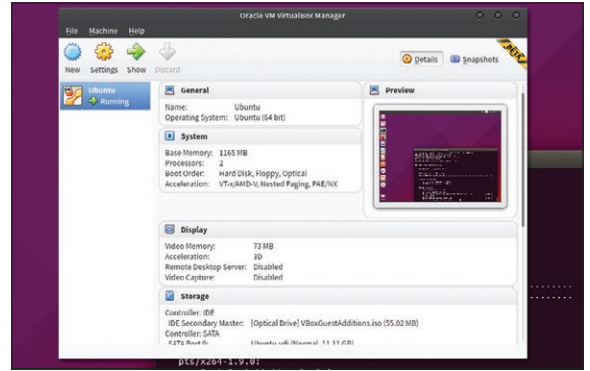
Эта бета-версия также предлагает разнообразные новые расширения набора инструкций для гостевых VM; добавляет поддержку устройств USB 3.0 в контроллере XHCI, поддержку drag-and-drop в двух направлениях для гостей и поддержку

шифрования образа диска. Есть также и ряд улучшений в пользовательском интерфейсе и поддержка экранов HiDPI, а также ряд долгожданных правок.

Но самая крутая функция версии 5.0 — это, конечно, поддержка паравиртуализации, доступ к которой можно получить в разделе System > Acceleration [Система > Ускорение] настроек виртуальной машины. *VirtualBox* предлагает несколько интерфейсов для паравиртуализации, таких, как Legacy, Minimal, Hyper-V (для Windows) и KVM (для Linux).

Паравиртуализация предлагает специально определенные стыковочные средства, которые позволяют гостевым системам получать доступ к частям оборудования хоста напрямую, повышая таким

«Крутая функция версии 5.0 — поддержка паравиртуализации.»



› Настройте гостевую ОС в *VirtualBox* с помощью самого последнего ISO Guest Additions ради наилучшего результата.

образом производительность. В наших тестах новая версия *VirtualBox* продемонстрировала значительно возросшую скорость гостевой ОС вместе с намного меньшим использованием CPU на стороне хоста.

Установка 5.0.0_BETA1 очень проста. Хотя по вышеприведенной ссылке доступен tar-архив исходника, Oracle предлагает большой набор прекомпилированных пакетов для самых популярных разновидностей Linux для предприятия и также некоторых систем рабочего стола, что весьма удобно.

Имеются готовые к установке пакеты для RHEL версий от 5 до 7, Fedora, openSUSE, SLES, RPM и Deb для Ubuntu и Debian, а также *VirtualBox Extension Pack* и *ISO Guest Additions* для версии 5.0.0 beta1.

Программа создания интеллект-карт

FreeMind

Версия: 1.0.1 Сайт: <http://freemind.sourceforge.net>

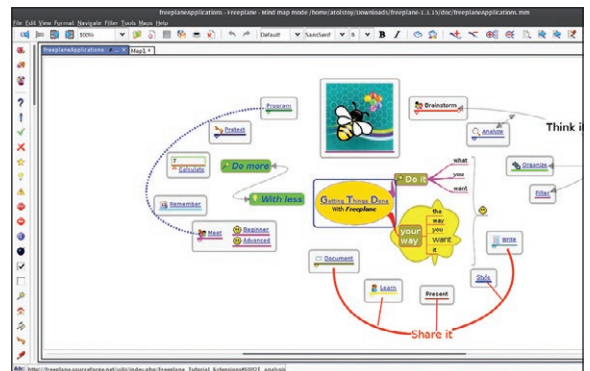
FreeMind — это специализированная программа для создания интеллект-карт, применяемых многими для персонального менеджмента. Как и другие подобные программы, *FreeMind* позволяет редактировать иерархический набор идей вокруг центральной концепции. При добавлении идей карта вырастает в нелинейную схему и помогает в организации мозговых атак в новых проектах.

Проект предлагает разные опции скачивания, но, вероятно, наилучшей будет скачать *freemindbin-max-1.0.1.zip* (универсальный архив), распаковать его и запустить в терминале *./freemind.sh*. Главное окно предлагает обширное рабочее пространство и массу крошечных значков на вертикальной и горизонтальной панели инструментов. Однако большая часть эффективной работы выполняется с помощью клавиш и секретных подсказок, узнать которые можно из документации *FreeMind*. Вот несколько примеров

основных действий. Карта всегда начинается с центральной концепции, которую можно перемещать, щелкая по пустому месту за пределами пузыря. Нажмите на Insert, чтобы создать дочерний узел, или на Enter, чтобы создать узел-брат под ним. или Shift+Enter для создания нового узла-брата выше. По щелчку правой кнопки на узле появится контекстное меню, где можно настроить узел, добавить в него один и более значков, форматировать шрифт ярлыка, добавить облако, изображение или гиперссылку к узлу, и т. д.

Приведенный в качестве примера рисунок (выше, справа) иллюстрирует разнообразие опций индивидуальной настройки, которые помогают создавать

«Настройки помогают создавать красочные и сложные карты.»



› Владейте искусством создания интеллект-карт с помощью множества расширенных функций во *FreeMind*.

красочные и сложные карты. Чтобы найти текст в узле и все его узлы-потомки, нажмите Ctrl+F или выберите Node > в контекстном меню узла. Пересмотрев свои цели, вы можете решить изменить расположение и порядок узлов. Для этого выберите узлы с помощью клавиш Ctrl или Shift и щелкните по нужному узлу и перетащите его в нужный узел. Чтобы создать графическую ссылку от одного узла к другому, перетащите узел с помощью правой кнопки мыши.

Опций во *FreeMind* куда больше. Документация предлагается рекурсивно в основном на Java формате *FreeMind*, и он потребует, чтобы ваш браузер разрешил использование плагина IcedTea. LXF

На диске

Дистрибутивы, приложения, игры, подкасты и всякое-разное...

Лучшее из Интернета, упакованное в 9 ГБ качественного DVD.



Дистрибутивы

Будучи отчасти помешанным на контроле, я люблю Gentoo, но его поддержка требует лишней работы, ведь системный администратор берет на себя часть функций по «уходу» за дистрибутивом. Каждая инсталляция Gentoo может быть уникальной, но подразумевает большие затраты времени: Gentoo позволяет вам создать свой дистрибутив, даже поощряет вас к тому, и это — хорошая платформа. Существует несколько известных дистрибутивов, чьи корни — в Gentoo, даже если об этом нечасто говорят, например, System Rescue CD и ChromeOS; а вот Sabayon ничуть данного факта не скрывает.

Возможность взять созданное другими, чтобы создать нечто лучшее, не только основа развития свободного ПО: это принцип, уходящий корнями в глубь столетий. Всем известно, как Ньютон сказал, что он стоит на плечах гигантов, однако эта фраза звучала и за сотни лет до него. То, что важные основные принципы скрывают за патентами и авторским правом, лишь тормозит прогресс ради краткосрочной эгоистичной выгоды. Проекты типа Sabayon и ряда респинов Ubuntu демонстрируют, как ценно дать другим применить ваш труд и сделать его лучше.

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.

Дроид на вашем рабочем столе

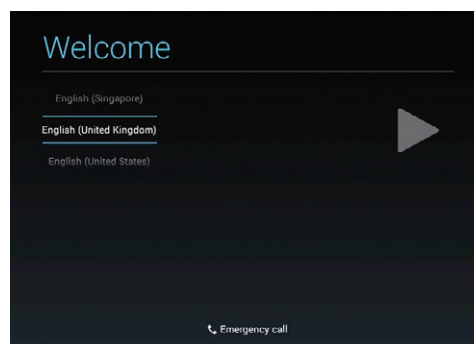
64-БИТНЫЙ

Android-x86 4.4-r2

Какой в наши дни самый популярный дистрибутив Linux? Возможно, вы назовете Ubuntu, если думаете о настольном варианте, или Debian — для сервера, или, может быть, Red Hat — для бизнеса; однако реальность такова, что огромную долю устройств, на которых работает Linux, сегодня составляют мобильные устройства. Может быть, Android и не GNU/Linux, но это Linux. Так что самое время включить в наш диск его версию. Не мобильную версию (она уже есть на вашем телефоне), а порт для рабочего стола.

Да-да, Android был портирован на стандартную архитектуру ПК в виде Android-X86, и у нас есть для вас версия 4.4 (Kit-Kat). Есть порт Lollipop, но он на текущий момент чутьком слишком экспериментальный, и вдобавок это не та система рабочего стола, которой вы захотите пользоваться ежедневно. Однако с ней интересно познакомиться, особенно если вы хотите писать программы для Android.

Android-X86 — это live CD, но его можно и установить. Он немного капризен насчет оборудования, так что вам, возможно, придется использовать опцию загрузки VESA, чтобы получить работоспособный дисплей; и не ждите, что он заработает на виртуальной машине. Одна из причин в том, что Android рассчитывает на беспроводное соединение с Интернетом, что и неудивительно, поскольку мы уже видели мобильный телефон с портом RJ-45! Но если ваше оборудование совместимо, будет очень интересно запустить



Android на ноутбуке, в том числе и с целью серьезных экспериментов.

Опция установки в меню загрузки предложит вам простейшую программу установки *Curses*. Для Android-X86 нужен один свободный раздел, и если вы его еще не создали, программа установки перенаправит вас в *cfdisk*, текстовый редактор разделов, для его создания, так что, возможно, вы решите изменить размер вашей существующей файловой системы и создать новый раздел в *GParted* или чем-то подобном, прежде чем загружать программу установки. Даже если вы хорошо знакомы и с рабочим столом Linux, и с Android, запуск Android-X86 на «нормальном» компьютере — это совершенно другое дело. К счастью, вы найдете большое количество документации и немало советов на www.android-x86.org.

Сделайте это Google

Вы больше не потеряете свои данные. Учетная запись Google содержит их в безопасности.

Получить учетную запись



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?

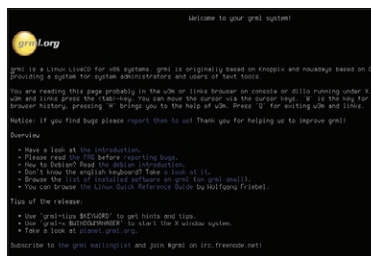
Ответы приводятся в [Index.html](#) на диске.

CD-супергерой для спасения системы

32-БИТНЫЙ

Grml 2014.11

Когда появится проблема, не угадаешь... и полезно всегда иметь под рукой диск аварийного восстановления системы. Grml — как раз такой диск: можно использовать ISO с DVD, или просто загрузить его с LXF DVD, если вам нужно аварийное восстановление системы. Это урезанная текстовая версия — слишком мало места оставило включение стольких Ubuntu; но в ней есть все инструменты, способные помочь, если у вас не загружается ОС, повреждена файловая система, забыт пароль или стряслась иная беда из тех, что случаются только с другими.



Мы наколдовали

32 и 64-БИТНЫЙ

Ubuntu 15.04

В этом месяце у нас двойной Ubuntu — один из наших традиционных ремиксов рабочих столов с Unity, KDE, LXDE, Xfce и Mate. DVD загружает рабочий стол Unity. Решив попробовать другой рабочий стол, выйдите из Unity, нажмите на значок с шестеренкой в верхней части экрана входа в систему для выбора другого рабочего стола, и затем войдите как пользователь ubuntu без пароля.

Live DVD всегда сначала загружает Unity, но после установки он запомнит ваш выбор для следующей

загрузки. Ремикс-версия существует только в 64-битном виде, но для тех, кому нужна 32-битная версия, мы включили стандартный 32-битный Ubuntu с Unity. Вы можете выбрать желаемое в меню загрузки DVD.

Примечание: с ремиксом рабочих столов имеется небольшая глюк. Рабочий стол Unity, загружаясь, открывает пустой документ *Abiword*; просто закройте его после загрузки. К сожалению, у нас не было времени на исправление: мы едва уложились в сроки, включая Ubuntu 15.04 в состав диска этого месяца. LXF



И еще!

Системные инструменты

Главное

Checkinstall Установка tar-архива с помощью менеджера пакетов.

GNU Core Utils Основные утилиты, обязанные быть в каждой операционной системе.

Hardinfo Инструмент тестирования системы.

Kernel Исходный код самого последнего стабильного релиза ядра.

Mentest86+ Проверьте ОЗУ на предмет сбоев.

Plop Простой менеджер загрузки для запуска ОС с CD, DVD и USB.

RaWrite Создавайте загрузочные диски в MS-DOS в Windows.

SBM Независимый от ОС менеджер загрузки с несложным интерфейсом.

Wvdial Соединяйтесь с Интернетом через телефонный модем.

Чтение

Книжная полка

Руководство по созданию скриптов Bourne Shell Начинать осваивать скрипты оболочки Bash.

Расширенное руководство по скриптам Bash Изучите написание скриптов еще лучше.

The Cathedral and the Bazaar [Собор и Базар] Классический текст Эрика С. Реймонда [Eric S Raymond], объясняющий преимущества открытой разработки.

Справочник администратора Debian Базовое руководство для системных администраторов.

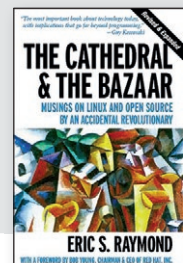
Введение в Linux Удобное руководство со множеством подсказок для начинающих пользователей Linux.

Словарь Linux Linux от А до Я.

Linux Kernel in a Nutshell [Ядро Linux в Двух Словах] Введение в ядро, написанное великим магистром ядра Греггом Кроа-Хартманом [Greg Kroah-Hartman].

Руководство системного администратора Linux Контролируйте свою систему

Обзор инструментов Полный обзор инструментов GNU.



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:




LXF195
Апрель 2015

- » Raspberry Pi 2 Взята планка 1 Гб ОЗУ
- » Языки скриптов Вот и поговорим
- » NoSQL Хватит SQLить, даешь инакомыслие
- » 1С для Linux Бухгалтеры, внимание!

LXFDVD: Fedora 21, Ubuntu 14.10, PCLinuxOS 2014.1, ArchBang 2015.01, OpenMediaVault, ExTiX 15.2, MakuluLinux 8.0, SalentoS 14.04.2, 11 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_195/
PDF-версия: shop.linuxformat.ru/elxf_195/




LXF196
Май 2015

- » 100 программ-жемчужин Без них вам не жить
- » Аноним Фигаро Дистрибутивы заматают следы
- » Торификация Упрячем свой трафик
- » Ядро Linux Позови меня, позови...

LXFDVD: Mint 17.1, Kodi 14, Scientific Linux, Tiny Core, руководство по Android, 11 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_196/
PDF-версия: shop.linuxformat.ru/elxf_196/



LXF197
Июнь 2015

- » Жизнь с Малиной Лучшие проекты для Raspberry Pi
- » Выбор виртуальной реальности Компьютер в компьютере
- » Lego для инженеров Моделирование систем
- » Пламенные звуки Аудио в Linux

LXFDVD: Bodhi 3.0, SteamOS Beta, Zentyal, Antergos, Fedora, Robolinux, руководство по хромбукам, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_197/
PDF-версия: shop.linuxformat.ru/elxf_197/

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

LINUX
ФОРМАТ

Страница 1

ДИСТРИБУТИВЫ

Ubuntu 15.04 (32- и 64-битные)
Android x86 (32-битный)
Erm1 201411 (32-битный)

НОТРЬСКС

Audacity Аудиоредактор и программа для записи
Brackets Текстовый редактор
Fzkoll Пакет Metasploit
FreeMind Программа создания интеллектуальных карт
Geany Почтовый клиент
Lipphone SIP-клиент
Netsint Web-браузер
Paspang Emacs Настольная игра
Shotwell Программа виртуализации
Vitalbox Программа виртуализации

СРАВНЕНИЕ КЛИЕНТОВ
УДАЛЕННЫХ ДЕСКТОПОВ

OpenOffice Desktop
NoMachine
Remmina
TigerVNC
X2Go

УЧЕБНИКИ

Scala
Systat
SystemTap

ПОМОЩЬ

FAQ DVD
Руководство новичка
Руководства

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!

Обнови

15.04

СВЕЖИЙ РЕЛИЗ

Новые для 2015!

- » Готов к работе
- » Готов к установке
- » Обновитесь сегодня!

32 & 64-битный

LINUX
ФОРМАТ
ПЛЮС: ANDROID 4.4 X86
ВСЕ, ЧТО ТРЕБУЕТСЯ ДЛЯ СТАРТА В LINUX

Содержание

LINUX FORMAT

Страна 2

ГЛАВНОЕ
Checkinstall
Coreutils
Hardinfo
Kernel
Memtest86+
Plop
SBM
Wvdial

The Cathedral and the 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 Краа-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
4MLinux 13.0 (32-битный, ISO-образ)
Antergos 2015.06.03 (32- и 64-битный, ISO-образ)
Chromixium OS 1.5 (32-битный, ISO-образ)
Linux Mint 17.2 MATE (64-битный, ISO-образ)

ДОКУМЕНТАЦИЯ: 10 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

Advanced Bash Scripting Guide Подробное руководство по программированию на Bash для начинающих
Bash Guide for Beginners Руководство по Bash для начинающих
 Bourne Shell Scripting Начальное руководство по программированию на Bash

**Пожадуйста, перестаньте использовать
 имя Аджного Диска ознакомьтесь
 с инструкцией, опубликованной
 в журнале на стр. 109!**

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕФЕКТИВНЫЕ 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 авторизованный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных в вашем жестком диске. 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, представленных на нем программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, с нами, провозгласившим себя «самым добрым из братьев».

Тираж издательства ООО «Марком», 198652, Россия, Ленинградская область, Всеволожский р-н, дер. Юрки, Школьная ул., 7-а. Лицензия ИПТР ВАО № 77-03.

Создание установочных дисков при помощи cdcrecord

Самый быстрый способ записать ISO-образ на чистую матрицу — это обратиться к программе *cdcrecord*. Для всех перечисленных ниже действий потребуются права суперпользователя-root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdcrecord -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:

```
cdcrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdcrecord*, 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

```
cdcrecord -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 нажимайте на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Использовать Linux для записи компакт-диска не обязательно. Программы вроде *cdcrecord* 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/>

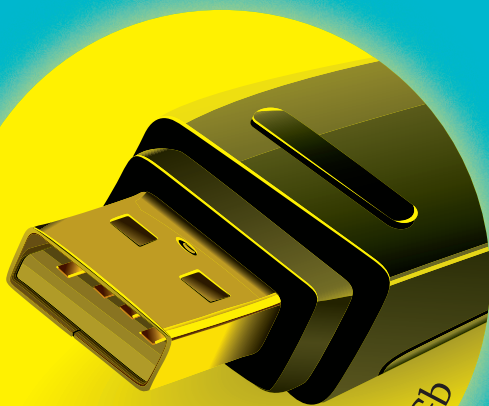
Отдел дистрибьюции
ГНУ/Линуксцентра
приглашает дилеров
и дистрибьюторов
к сотрудничеству!

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/

LINUX
FORMAT

Главное в мире Linux

Реклама в журнале
Linux Format

3-я обложка	200 000
2-я обложка	230 000
4-я обложка	275 000
Разворот	230 000
1/1 полоса (210×297 мм)	140 000
1/2 полосы (197×141, 92×280 мм)	75 000
1/4 полосы (92×113 мм)	45 000

Тел.: (812) 309 06 86

Цены указаны в рублях и не включают НДС (18%)

ГНУ/Линуксцентр
приглашает на работу!



ВАКАНСИЯ: Разработчик систем электронного документооборота

ТРЕБОВАНИЯ:

- » Опыт внедрения и сопровождения систем электронного документооборота на базе Alfresco
- » Умение описать бизнес-процесс
- » Знание SQL, понимание принципов построения и функционирования баз данных
- » Знание СЭД, отличных от Alfresco, приветствуется

ОБЯЗАННОСТИ:

- » Разработка систем с нуля, сопровождение
- » Настройка системы в соответствии с требованиями бизнеса
- » Написание инструкций для конечных пользователей

ПОДРОБНЕЕ: www.linuxcenter.ru/vacancy/

Футболки GNU/Linux

БОЛЕЕ 100 ВИДОВ. ВСЕ РАЗМЕРЫ И ЦВЕТА



WWW.LINUXCENTER.RU/SHOP/GIFTS/

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № Ф077-21973 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

Елена Ессяк, Даниил Кривошеин, Светлана Кривошеина, Валентин Развозжаев, Валерий Смирнов, Елена Толстякова

Редактор диска

Александр Баракин

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Андрей Смирнов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатель

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 12715

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@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Мэтт Билби [Matt Beilby], Джонни Бидвелл [Jonni Bidwell], Нейл Ботвик [Neil Botwick], Джоллон Браун [Jolynn Brown], Крис Браун [Chris Brown], Шон Конвей [Sean Conway], Мэтью Хансон [Matthew Hanson], Карл Ходж [Carl Hodge], Робин де Йонг [Robin de Jongh], Кевин Ли [Kevin Lee], Хуан Мартинес [Juan Martinez], Лес Паундер [Les Pounder], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Александр Толстой [Alexander Tolstoy], Михалис Цукалос [Mihalis Tsoukalos], Лиз Антон [Liz Upton], Евгений Балдин, Андрей Гондаренко, Андрей Прахов, 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, пр. Медиков, 5, корп. 7

Тел. +7 (812) 309-0686, e-mail: info@linuxformat.ru

По вопросам сотрудничества, партнерства, оптовых закупок:

partner@linuxcenter.ru

Авторские права: статьи, переведенные из английского издания Linux Format, являются собственностью или лицензированы Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на диски — CD или DVD, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственность за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].
GNU/Linux — заимствования 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+



В августовском номере

Соберем Linux-ПК

Настало время обновить вашу машину — и мы расскажем вам, как это сделать, а также всё, что надо знать пользователю Linux.

Исследуем Systemd

Systemd уже успел заразить дистрибутивы Ubuntu, Debian, Fedora и не только. Пора и вам болеть за него.

История времени

Поговорим об измерении времени операционной системой. Тут все непросто, это ж не календарь маяя.

Внутри IPv6

IP-адреса в мире кончаются — но не паникуйте! Переходите на IPv6, пока это круто и пока до него не добрались бомжи.

Содержание будущих выпусков может меняться — благодаря этому ускоряется прогресс.

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.

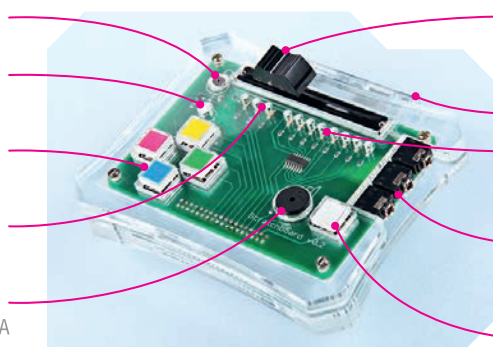
ДАТЧИК ЗВУКА*

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

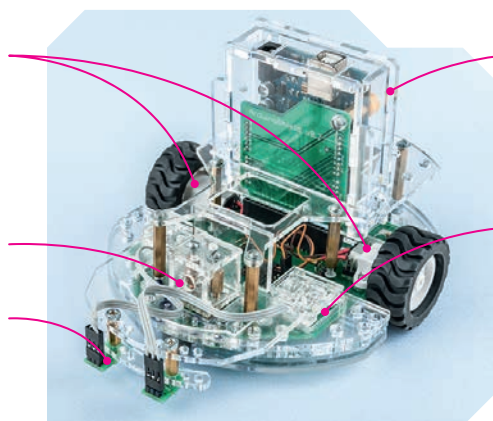
обеспечивает возможность:

- 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.