

DVD ДАРОМ!



UBUNTU STUDIO 16

LINUX FORMAT

Главное в мире Linux

Май 2017 № 5 (223)

PI ZERO W ПОДРОБНО

» Bluetooth » Wi-Fi » Всего за £ 10

Соберите классные устройства с новехоньким Pi и руководством для умельцев



Canonical 2017

Мы поговорили про дронов, IoT, роботов, телефоны и мировое господство!



Открытая демократия

« Я хотел бы быть в системе, где нет партий. Я хотел бы быть в системе, где есть представители »

Джеймс Смит — о том, как Open Source может изменить мир

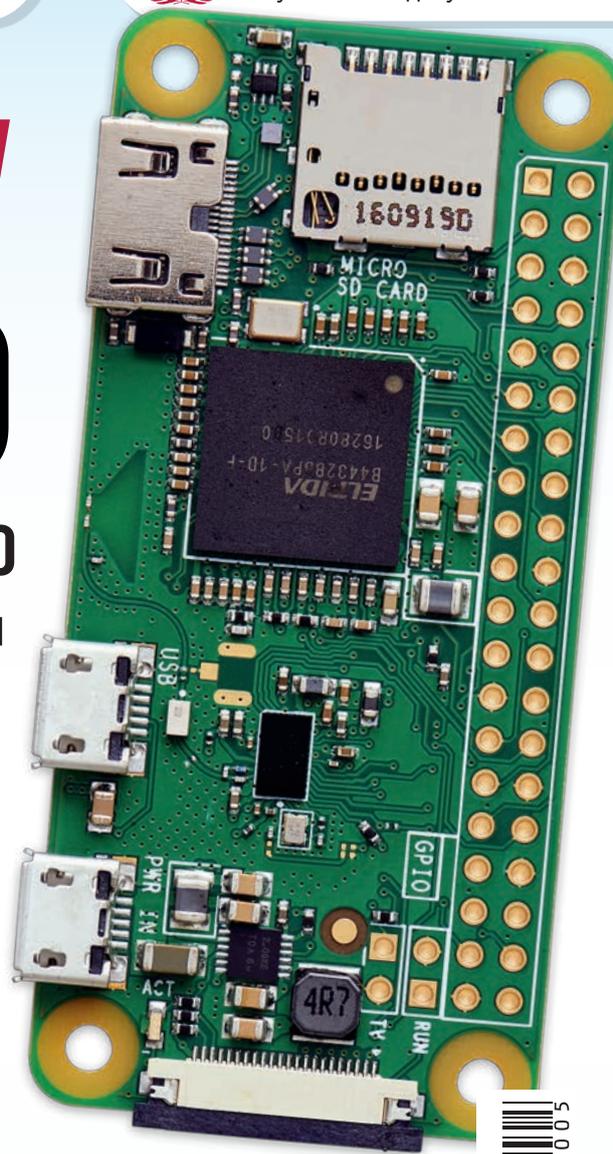


72 страницы учебников и статей!

- » Гоняем Blu-ray в Linux
- » Шифруем файлы внутри картинок
- » Шлифуем сервер с Varnish



Академия кодига: Пишем в Python команды-утилиты



AMD Ryzen 1800X CPU

16 потоков силици — и зачем вам уже Intel?

Мобильные приложения

Android Studio

» Начните разработку своего приложения прямо сейчас!



Сравнение

Открытые CAD

» Станьте лучшим мастером с идеальными инструментами



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: в этом номере мы рассматриваем Pi Zero W. Какие у вас планы на Pi в этом году, или какой новый проект для Pi вы надеетесь увидеть?



Джонни Бидвелл

Мне очень хотелось бы соорудить на Raspberry Pi рабочую станцию аудио, воспользовавшись интерфейсом pi4sound с Indiegogo. Но поскольку я ничего не понимаю в создании музыки, да и музыкальные вкусы у меня довольно спорные, уж лучше я заставлю работать Netflix в Raspbian.



Нейл Ботвик

Я использую парочку Raspberry Pi, чтобы сконструировать самодвижущийся автомобиль. Проблемы еще остаются — в основном, как избежать наезда на кошку! Как только я переделаю мой секретный подземный бункер, оснастив его Pi, займусь этим вплотную.



Нейт Дрейк

Пока я это пишу, Pi Zero W радостно летит ко мне в почтовый ящик. Моим первым проектом будет создание шифровальной машины Purple, которую японцы использовали во время Второй мировой войны, с применением эмулятора Python (это не японцы его применяли, а я буду). Потому что нет лучше технологий, чем ретротехнологии.



Джон Найт

Я был в восторге от Pandora — карманного ПК на Linux, с подходящей клавиатурой и эпическим сроком работы батареи, но такие штуки практически невозможно купить! Я знаю, что уже существуют ноутбуки на Pi, но тоскую по магии и миниатюрному форм-фактору Pandora, а Zero мог бы стать отличной базой для новой альтернативы.



Лес Паундер

Микроскопические размеры и цена Pi Zero W делают его идеальным кандидатом для носимых проектов. Скажем, создания одежды, реагирующей на внешние данные — типа велосипедной куртки, которая следит за освещенностью и включает светодиоды для вашей безопасности. Или регистрирует загрязненность и предупреждает вас о мерах предосторожности.



Валентин Синецын

Мои планы — раскопать в ящике самый первый Raspberry Pi образца 2012 года, купить, наконец, патч-корд, поставить на него XVMC... то есть OpenLEC 8.0, и наслаждаться тем, что прогрессивное человечество имеет уже лет пять как.



Кризис «перепроизводства лишнего»

» Глядя на Raspberry Pi Zero, невольно задумываешься о торжестве идеи System-on-Chip. Одна микросхема (в Zero W — две) и немного пассивных компонентов собраны на печатной плате минимально необходимого для размещения требуемых разъемов размера. Ничего лишнего, но свои задачи устройство решает блестяще. Лет 30 назад я так же смотрел на персональный компьютер: и эта штука из десятка микросхем позволяет делать почти всё то же, что и ЕС ЭВМ размером с комнату?!

По моему мнению, лаконичность решений стала одной из причин успеха команды Raspberry Pi. Инженеры красиво решили задачу, сформулированную в технических, а не маркетинговых терминах. А еще у стратегов некоммерческого Raspberry Pi Foundation по определению нет совета акционеров и не стоит задача убеждать в своей правоте бизнес-аналитиков. Можно не торопясь создавать что-то действительно нужное людям. И никто не потребует добавить ту или иную функцию только затем, чтобы отличаться от конкурентов, и не позже, чем через три месяца... а иначе Уолл-стрит отреагирует...

В ИТ, как и во многих других областях человеческой деятельности, успех зависит от постановки задачи. Если люди берутся делать дело, получается Raspberry Pi. А если целью является исключительно повышение капитализации, получается... ну, например, Windows Vista.

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

«Мы строим слишком много стен и недостаточно мостов.» Исаак Ньютон

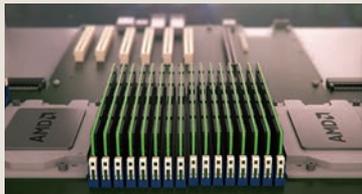
Обзоры

AMD Ryzen 7 1800X 12

Запускаем процессор с совершенно новой архитектурой от AMD, выясняя, на что способна эта 8-ядерная, 16-поточная сила, а заодно и весь модельный ряд.

AMD Ryzen и Linux 14

Берем новейшее ядро Linux и чипсет Ryzen X370, и сравниваем их с целью увидеть, что поломается. Надеемся, что не испытательный стенд LXF...



➤ Райзен — рай. Райзен — это резонно. Райзен неутраивим...

Dell XPS 13 9360 16

Ноутбук с Linux, которым Dell покорила мир, вернулся! Освеженный новейшим Kaby Lake от Intel, гордый всё тем же ультравысоким разрешением экрана, а теперь еще и с USB C для всяких там портов.



➤ Знаменитый ноутбук с Linux отныне снабжен USB C.

Hitman 17

Сами решайте, как выполнить миссию: хирургически удалив ключевых вражин или разгромив всё подряд оружием, бомбами либо чуелом лося.

Torment: Tides of Numenera ... 18

Создатели этой игры видят будущее Земли не иначе как пост-апокалиптическим. Torment значит «мучение»...

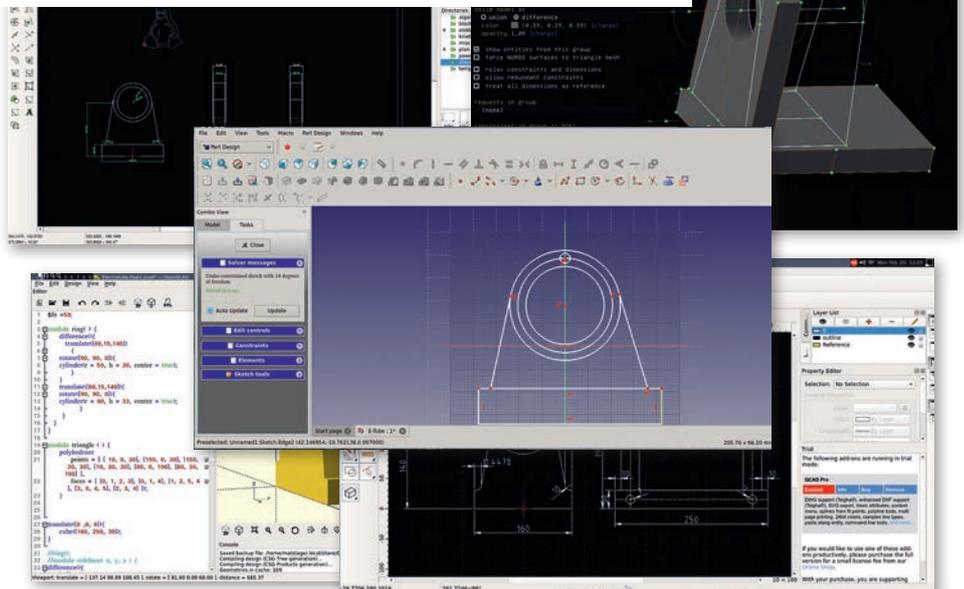
Grim Fandango 19

Детектив и история любви в одной посуде, и притом развертывающиеся в загробном мире древних индейцев.

Лучшие устройства IoT на Pi

Вооружитесь лучшим Pi на сегодняшний день и соберите лучшие устройства Интернета Вещей, следуя нашему руководству для начинающих. Оно на стр. 30

Сравнение: Открытые программы CAD с. 24



На вашем бесплатном DVD



Ubuntu Studio 16.04 64-битный openSUSE Tumbleweed 64-битный XenialDog 2017 Xfce 32-битный

» Каждый месяц — только лучшие дистрибутивы
ПЛЮС: HotPicks, Сравнение, и не только! **с. 106**

Побалуйте себя и любимых подпиской на LXF!



Доступно в AppStore!



www.linuxformat.ru/subscribe

Ищите в номере

Tizen как родной 22
Мы добрались до написания нативных приложений! Обновите-ка свои знания языка C.

Blu-ray в Linux 46
Исследуем, как пользователям Linux насладиться просмотром фильмов Blu-ray... и обнаруживаем жуткие инструменты и DRM, которые мы вынуждены обходить.



Интервью

«Главное, что принесло движение Open Source — сотрудничество»



Джеймс Смит рассуждает про открытую демократию **с. 40**

Учебники

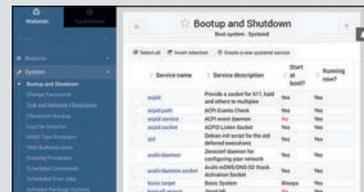
Шифрование
Outguess 54
Скройте свои личные данные в обычных файлах с помощью прекрасной стеганографической программы.

Виртуальные машины
Proxmox 58
Делаем первые шаги по виртуализации серверов, на платформе виртуализации Proxmox.



» Мы и в своей-то реальности не уверены...

Серия о серверах
Samba и Webmin 62
Базовый сервер LXF мы настроили, а теперь настроим Webmin и Samba себе на радость.



» Послушаем дружка, загрузим и отключим.

Безопасность
Ключи GnuPG 66
Ваш ключ GnuPG — едва ли не самое дорогое, что у вас есть. Сделаем его надежным и будем держать в тайне.

Web-серверы
Кэш Varnish 70
Устанавливаем и настраиваем HTTP-кэш Varnish, чтобы ваши сайты никогда не падали. По возможности.

Интернет вещей
Arduino и Apache 74
Подключаем радиомодуль к Arduino Uno, а Apache — к управлению процессом: пусть делает черновую работу за нас.

Игрострой
Cocos2d 78
Напишем игру, пользуясь удобным движком, который богат методами взаимодействия объектов.

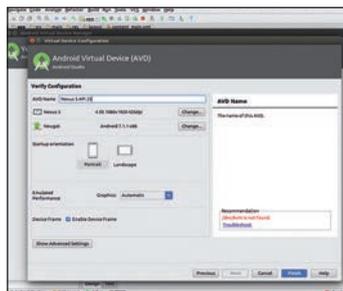
Академия кодига

Android Studio 82
Сделайте первые шаги в разработке приложений для Android! Демистифицируем основы, и вы тут же начнете клепать приложения и отправлять их в Google Play.

Рисуем в D3.js 92
Объезжаем на кривой библиотеку D3.js, чтобы генерировать любые графики по сложным наборам данных.

find 88
Эта утилита командной строки невероятно полезна для быстрого поиска файлов и каталогов.

Фракталы 96
Отправляемся за открытиями бесконечной красоты математики.



Постоянные рубрики

Новости 6
AMD щегольнула новым графическим процессором, Ubuntu простилась с Unity, Zbook обскакал Mac Pro, электроника станет плоской, CuneiForm вновь среди лидеров OCR, выпала ROSA R9, пишется встраиваемая ОС QNX, а суперкомпьютер DGX-1 от Nvidia уместился на столе.

Вести мобильных ОС 20
Lenovo предлагает планшет-трансформер, BlackBerry и TCL поставили на телефон аппаратную клавиатуру, автомобили обзаводятся «зрением», а HP обогнала Microsoft.

Сравнение 24
Мы наш, мы дивный новый мир построим — используя лучшие инструменты CAD с открытым кодом: FreeCAD, LibreCAD, OpenSCAD, QCAD, SolveSpace.

Интервью LXF 40, 50
Слушаем рассказы Джеймса Смита о партии открытого кода и Майка Белла о Canonical на MWC-2017.

HotPicks 100
Отведаете горяченького! Лучшие в мире новинки свободного ПО: 404 Elm Street, Argos, AzPainter, Buttercup, Cloudcmd, Findimagedupes,

GRSync, KSnip, Latte Dock, Machines vs Machines, QRab.

Диск Linux Format 106
Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108
Ощутите поток СПО сквозь вас и слейтесь с терминалом, как в LXF222 — именно этого и хотел бы Джа-Джа.

Через месяц 112
Ubuntu приходит только раз в году (вообще-то два раза, но не будем считать), и вот теперь он здесь и принес много-много радости... и слез.

В ЭТОМ НОМЕРЕ: Рывок графики » Новый поворот » Не только для игр » Предтеча из Вены » Распознаем всё » И вновь ROSA » ОС реального времени » Nvidia — супер

ГРАФИЧЕСКИЕ ПРОЦЕССОРЫ

Даешь 8K!

Компания AMD показала возможности своего GPU следующего поколения.

На Международной выставке NAB Show LA 2017 в Лас-Вегасе (24–27 апреля), целевой аудиторией которой является теле- и киноиндустрия, где использование 8K — растущая и неизбежная тенденция, возможности нового GPU Vega от компании AMD демонстрировались на примере обработки видео 8K в редакторе Adobe Premiere Pro CC 2017 и постобработки 4K на движке рендеринга Radeon ProRender. GPU Vega первоначально был показан на выставке CES в январе, а затем на проведенном AMD в феврале Технологическом дне; теперь же производитель раскрыл множество технических деталей. Vega использует память HBM2 и новый движок рендеринга графики, контроллер кэша с высокой пропускной способностью,

новые программируемые геометрические конвейеры и вычислительные блоки следующего поколения; реализована настройка на искусственный интеллект с возможностью выполнения 8-битных операций с плавающей точкой, что критически важно для машинного обучения. В ходе выставки AMD также объявила о наличии планов по созданию серверного мегачипа, в котором высококачественный серверный чип будет объединен с GPU Vega, и представила обновленную видеокарту Radeon Pro Duo с двумя GPU Polaris 10, соединенных через CrossFire.

Поставки первых GPU Vega ожидаются во втором квартале; вероятно, в ближайшее время будут выпущены и графические карты с этим GPU (Radeon RX Vega может

быть анонсирована на выставке Computex 2017, май — июнь 2017 г.). Контент 4K уже стал обыденностью для пользователей компьютеров; через 2–3 года таким же привычным будет разрешение 8K. Ранее компания Dell заявила, что ее монитор 8K работает с GPU Radeon WX 7100 на Polaris, однако производительность не оптимальна.

» Компания AMD официально подтвердила: выпуск графических процессоров Vega начнется во втором квартале 2017 г.



» Рубрику готовил
**АНДРЕЙ
ГОНДАРЕНКОВ**

СМЕНА КУРСА

Жизнь после Unity

Пользователям придется сменить редакцию Ubuntu — или дистрибутив.

В июне Canonical полностью прекратит выпуск обновлений для смартфонов на платформе Ubuntu Phone, разработчики лишатся возможности загружать свои приложения в Ubuntu Store, продажи платных приложений в нем прекратятся, а к концу года Ubuntu Store будет закрыт; перед нами один из самых агрессивных из когда-либо существовавших графиков завершения популярного проекта.

Тем временем, на Ubuntu Wiki опубликован график подготовки релиза Ubuntu 17.10, получившего кодовое имя Artful Aardvark. Дата финального выпуска намечена

на 19 октября; альфа-релиз всех официальных редакций семейства *Ubuntu состоится 29 июня, заморозка функций — 24 августа, бета-выпуск — 31 августа, заморозка ядра — 5 октября. Ubuntu 17.10 предшествует очередному релизу LTS, поэтому внедряемые в него изменения традиционно наиболее амбициозны. Интерес к этому выпуску подогревают отказ от Unity 7 в пользу Gnome Shell и вероятность использования Wayland по умолчанию; в данном цикле мы также увидим интеграцию GCC 7, Mesa 17.2 или, возможно, Mesa 17.3, ядро Linux 4.13–4.14.

Итак, редакция Gnome становится в Ubuntu основной, пользователям предоставят инструменты для плавного перехода с Ubuntu 16.04 LTS Unity на Ubuntu 18.04 LTS Gnome, а пакеты с Unity 7 войдут в состав Ubuntu 18.04 LTS в качестве опции, в репозитории Universe (без официальной поддержки). Видимо, Марк Шаттлворт [Mark Shuttleworth] решил не пытаться «объять необъятное» и сосредоточился на бизнес-решениях и «десктопе для людей»; а вот пользователям Ubuntu Phone и настольных персоналок с Unity теперь необходимо делать новый выбор.

НОУТБУКИ

Мобильные рабочие станции

Ноутбуки HP Zbook во многом превосходят стареющий Apple Mac Pro.

Рабочая станция Mac Pro выпущена Apple в 2013 г. и за истекшие 4 года обновлялась лишь незначительно; серьезные обновления производитель обещает в 2018 или 2019 г. Такая политика Apple вынуждает пользователей уйти на ПК с Linux или Windows: уже даже мобильные рабочие станции PC догоняют Mac Pro по функциональности и производительности, а настольные рабочие станции вроде HP Z и десктопы Dell Precision значительно превосходят Mac Pro по технологиям.

Новые ноутбуки HP Zbook оснащены портами Thunderbolt 3, памятью DDR4, процессорами Kaby Lake и Xeon от Intel, а также новейшими GPU от Nvidia и AMD. Для сравнения, Mac Pro комплектуется портами Thunderbolt 2, старыми GPU от AMD, памятью DDR3 и процессорами Intel Xeon на базе архитектуры Ivy Bridge, которые были выпущены в 2013 г. Технические характеристики

Zbook'ов позволяют использовать их для создания контента виртуальной реальности или редактирования видео, где ранее был бы необходим Mac Pro. Ноутбуки Zbook поставляются с экранами 14", 15,6" и 17,3", их цена вдвое ниже, чем у Mac Pro.

ZBook 17 — самая мощная из новых рабочих станций HP — комплектуется накопителем HDD или SSD емкостью до 4 ТБ, 64 ГБ ОЗУ, процессором Intel Xeon E3 v6 или Core i5/i7 и двумя портами Thunderbolt 3, с поддержкой 2-х дисплеев 4K каждый. В качестве GPU может быть установлен новейший Nvidia Quadro P5000, позволяющий создавать контент VR. В комплектацию могут входить экран 4K, порты HDMI и USB 3.0; цена начинается от \$1519. Zbook 15 комплектуется накопителем до 3 ТБ и поддерживает GPU Nvidia Quadro M2200 (но не P5000); для него также предусмотрена конфигурация с экраном 4K, цена — от \$1419. Zbook

Пользователи, выполняющие ресурсоемкие задачи не только в офисе, по достоинству оценят возможности новой линейки мобильных рабочих станций HP Zbook.



Studio стоит от \$1399; эта модель похожа на Zbook 15, но гораздо тоньше, легче почти на 2 кг и имеет вдвое меньшие объемы памяти и накопителя; с учетом размера, возможна установка только GPU Nvidia Quadro M1200M; есть порт DisplayPort 1.2. Перечисленные модели поддерживают 802.11ac Wi-Fi и могут комплектоваться GPU AMD Fire-Pro. HP также представила (не назвав пока цену) ZBook 14u с экраном 14", на нем будут только процессоры Intel Core i5/i7.

НОВЫЕ ТЕХНОЛОГИИ

Гибкий микропроцессор

Микропроцессор с четырьмя командами стал предтечей «плоской» электроники.

Ученые Венского технического университета представили примитивный микропроцессор из графеноподобного двумерного материала. В процессоре всего 115 транзисторов, он обрабатывает по одному биту данных за один такт, использует набор из 4-х команд (NOP, LDA, AND и OR), схемотехнику техпроцесса 2 мкм, и не нацелен на превышение каких-либо эталонных рейтингов; пока, по словам его создателей, сделан «первый шаг на пути развития микропроцессоров на основе 2D-полупроводников».

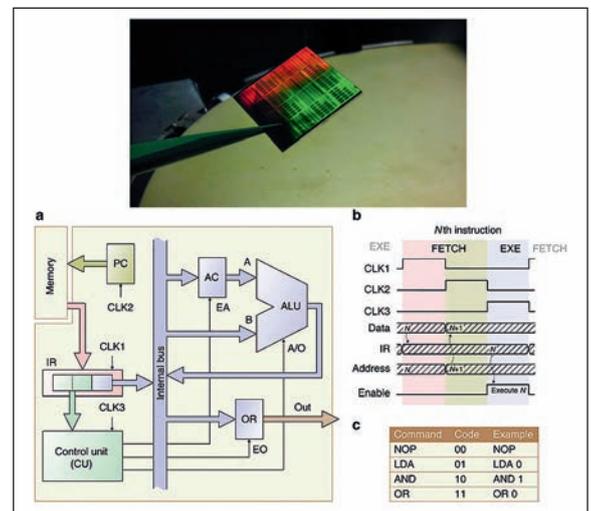
В отличие от современных полупроводников, действие которых основано на трехмерных физических свойствах материалов, из которых они изготовлены, 2D-материалы, такие как графен или ди-халькогениды переходных металлов (TMD), действительно двумерные, «плоские»: они выполнены из кристаллов, толщина которых составляет всего один слой атомов, что позволяет их сгибать. TMD представляют собой соединения, состоящие из переходного металла (молибден или вольфрам)

и халькогена (сера, селен или теллур), и образуют слои, подобно графену, но, в отличие от последнего, являющегося проводником, TMD — полупроводники. Штефан Вахтер [Stefan Wachter], Дмитрий Полюшкин и Томас Мюллер [Thomas Müller] из Института фотоники и Оле Бетге [Ole Bethge] из Института твердотельной электроники применили пленку дисульфида молибдена толщиной 0,6 нм, 2 слоя которой были нанесены на кремниевую подложку и разделены слоем оксида алюминия (причем подложка нужна лишь как несущая среда, а потому кремний можно заменить стеклом или другим материалом, включая гибкий).

Поскольку для производства «гибкой» электроники используются те же методы, что и для обычной, переход к элементам и транзисторам с длиной канала 100–200 нм не вызовет затруднений. Однако массовое производство таких чипов пока невозможно из-за отсутствия технологии производства высококачественной пленки дисульфида молибдена: работоспособными, из-за внутренних дефектов

пленки, являются лишь 5% транзисторов. Кроме того, необходим переход от технологии NMOS к менее энергоемкой CMOS, что потребует использовать иной 2D-полупроводник; одним из подходящих кандидатов является диселенид вольфрама.

Гибкий микропроцессор, созданный в Венском техническом университете, и его архитектура.



ОТКРЫТОЕ ПО

Маска, я вас... распознал!

Система распознавания *CuneiForm* вновь вошла в список лучших OCR мира.

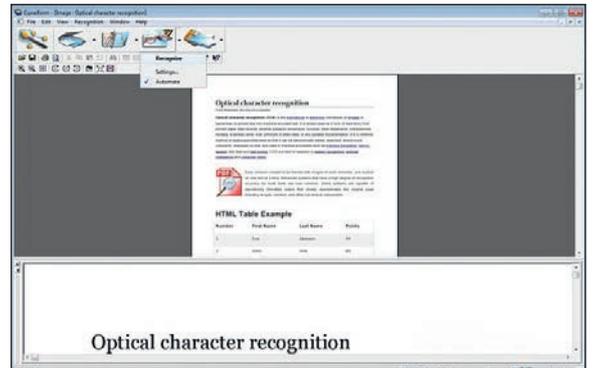
Открытая система оптического распознавания символов *CuneiForm* от компании Cognitive Technologies вошла в список лучших OCR-систем мира за 2017 г. (результаты исследования опубликовал международный портал Win-Beginner). Напомним, что в 2008 г. Cognitive Technologies открыла исходные коды *CuneiForm*; в том же году сообщество Open Source была выпущена версия Open OCR *CuneiForm* для ОС Linux; сообщество регулярно адаптирует *CuneiForm* для работы с новыми версиями ОС и иными программными ресурсами.

Возможности открытой OCR *CuneiForm*:

- » распознавание текстов на русском, английском, смешанном русско-английском, украинском, немецком, французском, испанском, итальянском, шведском и других (всего 26) языках;
- » работа в режиме автофрагментации для поиска текстовых блоков, таблиц и изображений, а также мощное средство ручной и полуавтоматической фрагментации;

- » распознавание таблиц любой структуры и сложности, в т.ч. и без линий графления;
- » автоматическое сохранение иллюстраций (черно-белых и цветных) и таблиц в получаемом на выходе документе;
- » полное сохранение топологии страницы;
- » поддержка пакетного режима сканирования и распознавания;
- » встроенный текстовый редактор для работы с распознанным текстом;
- » совмещенный показ изображений и результатов распознавания.

В системе используется целый ряд уникальных технологий, среди которых адаптивное распознавание, нейронные сети, когнитивный анализ альтернатив распознавания и др. OCR *CuneiForm* может распознавать любые полиграфические, машинописные гарнитуры всех начертаний и шрифты, получаемые с принтеров, за исключением декоративных и рукописных. В систему встроены специальные алгоритмы для распознавания текста с матричного принтера, плохих ксерокопий факсов и машинописи.



» Открытая OCR *CuneiForm* поддерживает ввод информации для распознавания со сканера или из графического файла, но не работает с документами формата PDF.

Система *CuneiForm* поставлялась в комплекте со сканерами известных брендов в Европе и мире, со всеми сканерами Hewlett-Packard для РФ, а также сканерами и МФУ Canon, Epson, Xerox, Samsung, Olivetti и др.; ее модуль распознавания встроено в пакет *Corel Draw*. В конце 1990-х Cognitive Technologies изменила стратегию разработок, сместив акцент на крупные проекты и сохранив распознавание как утилиту.

РОССИЙСКИЕ ОС

Роса оранжевого часа...

Вышел ROSA R9 — первый релиз новой платформы.

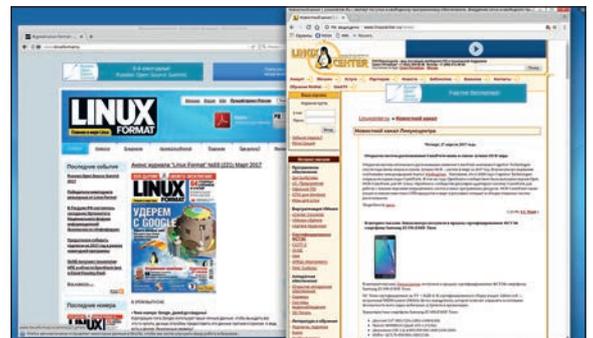
НТЦ ИТ РОСА представил ROSA FRESH R9 — первый релиз новой платформы 2016.1. Доступны 2 окружения рабочего стола — Plasma и KDE4, позже сообществом будут подготовлены образы с Gnome 3, LXQt и Mate. В соответствии с принятой политикой обновлений, этот релиз будет поддерживаться 4 года.

Среди изменений по сравнению с предыдущим релизом (R8):

- » в репозитории добавлены ldc (компилятор языка D), meson (высокопроизводительная система сборки) и другие новые пакеты;
- » в glibc добавлена заплатка, значительно повышающая скорость загрузки динамически разделяемых объектов (DSO);
- » в RPM добавлен экспериментальный генератор Provides для QML-модулей;
- » используется ядро 4.9.20 LTS, что позволяет устанавливать ROSA FRESH R9 в т.ч.

и на компьютеры с процессорами Intel Skylake последних серий;

- » используется Mesa 17.0.3 с поддержкой OpenGL вплоть до 4.5 и повышенной производительностью, что позволяет запускать игры на свободных драйверах. Также в Mesa включена поддержка нового графического API Vulkan;
- » проприетарные драйверы для видеокарт Nvidia используют внешние библиотеки GLVND вместо встроенных;
- » в репозитории для 64-битных систем добавлен проприетарный драйвер для видеокарт AMD — AMDGPU PRO (на данный момент — как экспериментальный). Проприетарный драйвер fglrx больше не поддерживается;
- » в X11 работа с устройствами ввода выполняется с помощью libinput;
- » выполнена реорганизация репозитория, некоторые популярные пакеты перенесены в репозиторий Main, что позволит повысить



» Многие опытные пользователи утверждают, что у ROSA Fresh одна из лучших реализаций KDE среди всех дистрибутивов с этим рабочим окружением.

качество их тестирования при обновлении версий;

- » в FFmpeg включена поддержка OpenCL;
- » Terminix (теперь уже tilix) собирается из исходников для 2 архитектур (раньше он являлся бинарным репаком для одной архитектуры);
- » в репозитории добавлен chrome-gnome-shell для расширений Gnome 3.

МНОГОЛИКИЙ UNIX

Мобилизуя потенциал

Энтузиаст создает рабочий стол для QNX 7.

Операционная система реального времени QNX предназначена главным образом для встраиваемых решений (автомобили, медицина, оборона и т. п.), однако сотрудник компании BlackBerry Элад Лахав [Elad Lahav] в свободное от основной работы время написал для QNX 7 экспериментальное рабочее окружение на основе Qt5, где каждое приложение можно запускать отдельно, в полноэкранном режиме, и с помощью QNX Software Development Platform (SDP) 7.0 портировал некоторое программное обеспечение.

В частности, это Vim, необходимый для написания кода разрабатываемого ПО; Subversion и SVN с сопутствующими библиотеками APR, APR Util и libserf; библиотеку Simple DirectMedia Layer (SDL), используемую многими играми с открытым

кодом. SDP 7 не ориентирован на рынок настольных компьютеров, однако он содержит необходимые стандартные блоки для создания диспетчера рабочего стола, в том числе композитного менеджера окон, стеков USB и HID для обработки устройств ввода, и библиотеки для отрисовки изображений и рендеринга шрифтов.

В качестве аппаратной составляющей была использована плата ASRock IMB-151 mini ITX с 4-ядерным процессором Intel Celeron.

Напомним, что в QNX 6 и более ранних версиях этой микроядерной ОС существовала официальная поддержка среды рабочего стола, но из выпущенного в марте этого года релиза QNX 7 такую опцию убрали. При этом QNX 7.0 впервые получила поддержку 64-битных платформ Intel x86 / ARM и aarch64, а также поддержку C++ 14.

СУПЕРКОМПЬЮТЕРЫ

Один против 400

В III квартале Nvidia начнет поставки суперкомпьютера DGX-1.

Мощный суперкомпьютер Nvidia DGX-1 с GPU архитектуры Volta, размеры которого позволяют установить его на обычном письменном столе, сумеет заменить собой несколько стоек с 400 серверами. Большую часть вычислительной мощности этого суперкомпьютера, который выглядит как обычный rack-сервер, обеспечивают 8 графических процессоров Tesla V100. Этот первый GPU новой архитектуры Volta Nvidia представила 10 мая. CEO компании Хуан Жэньсюнь [Jen-Hsun Huang] назвал цену DGX-1 с процессорами Tesla V100 — \$149 000.

У нового суперкомпьютера 40 960 ядер CUDA, эквивалентных мощности 800 CPU. Он заменит предыдущий DGX-1 на архитектуре Pascal, с мощностью 250 двухsocketных серверов. Система обеспечивает производительность 960 терафлоп с половинной точностью — 16-bit с плавающей точкой, и предназначена для задач машинного обучения. Помимо GPU, DGX-1 комплектуется двумя 20-ядерными процессорами Intel Xeon E5-2698 v4s с тактовой частотой



► Хуан Жэньсюнь, CEO Nvidia, представляет новую версию суперкомпьютера DGX-1 в ходе GPU Technology Conference в Сан-Хосе (Калифорния).

2,2 ГГц; система имеет 4 SSD емкостью 1,92 ТБ, работает на Ubuntu Linux и потребляет 3,2 кВт. GPU Tesla V100, установленный в DGX-1, в 5 раз быстрее процессоров архитектуры Pascal; он использует ряд новых технологий, в т. ч. высокопроизводительную шину NVLink 2.0 с пропускной способностью до 300 Гбит/с, включает более 21 млрд транзисторов и 5120 ядер, и комплектуется памятью HBM2 с пропускной способностью 900 Гбайт/с. Одновременно была представлена DGX Station, уменьшенная версия нового DGX-1. Она комплектуется четырьмя GPU Tesla V100, ее цена \$69 000, поставки также начнутся в Q3. LXF

Новости короткой строкой

► ООО «Национальный центр информатизации» разработало ОС «ОСь» на базе Linux, пригодную для работы с информацией, составляющей гостайну. Источник: izvestia.ru

► Google выплатит более 439 млн руб. штрафов и предоставит альтернативным поисковикам, в т. ч. инициатору разбирательства «Яндексу», новую возможность продвижения. Источник: www.kommersant.ru

► Компания «Кворум» переводит решение «СМЭВ-Интегратор» для информационного обмена между банком и сервисами госсистемы межведомственного взаимодействия на открытую СУБД PostgreSQL. Источник: www.quorum.ru

► Intel начала продажи «невероятно быстрого» накопителя Optane SSD DC P4800X, основанного на памяти 3D XPoint, емкостью 375 ГБ. Источник: www.cnews.ru

► В 2017 г. Huawei потратит \$3 млн на реализацию совместных проектов в РФ, конференцию в Москве и серию мероприятий в семи городах РФ. Источник: www.finanz.ru

► В ядре Linux 4.11 поддерживаются журналирование в программном RAID 4/5/6, новый тип сокетов SMC-R, протокол Opal для работы с самошифруемыми накопителями, отдельные буферы консолей с историей прокрутки. Источник: lkm.l.org

► Минкомсвязи рассматривает введение компенсационного платежа от госкомпаний, закупающих зарубежное ПО при наличии отечественного аналога. Источник: www.interfax.ru

► Вышли Debian 8.8 Jessie и второй релиз-кандидат Devuan Jessie 1.0 (ветка Debian без systemd). Источники: www.debian.org и lists.dyne.org

► Организация Open Invention Network включила в программу защиты Linux от патентных претензий 395 новых пакетов. Источник: www.openinventionnetwork.com

► Компания NAVITEL открыла доступ к API Навител Навигатор; теперь сторонние разработчики на ОС Android смогут задействовать его возможности в своих продуктах, в т. ч. коммерческих. Источник: navitel.ru

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



Новинки программного и аппаратного обеспечения в описании наших экспертов



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

Ubuntu без... Ubuntu?

Марк Шаттлворт 5 апреля 2017 г. объявил, что развитие рабочей среды Unity 8 и сервера отображения Mir прекращается. И «долгоиграющий» (LTS) релиз Ubuntu — 18.04 — с Gnome 3 по умолчанию. По значимости это превосходит переход с *Upstart* на *Systemd*, свершившийся пару лет назад: с системой инициализации большинство применителей имеют дело довольно редко, рабочая же среда перед глазами постоянно. Не секрет, что отношение к Unity, с момента его появления, было неоднозначным. Но и Gnome 3 не похвалится всеобщей любовью: у обоих десктопов «нетрадиционный» интерфейс, и смена одного на другой вполне описывается поговоркой «сменить шило на мыло». Почему упомянутое событие и обсуждалось столь бурно. Меньше внимания привлекло другое сообщение Марка, от 8 апреля, насчет деталей этого перехода. Хотя его значимость куда выше. Если первое сравнимо с отказом Папы Римского от поправки Filioque, то второе равносильно его переходу к атеизму. Ибо сводится к тому, что фирма Canonical по сути самоустраняется от развития настольного Ubuntu, а роль «головного» десктопа переходит к Gnome, ранее бывшего «бедным родственником». О последствиях можно лишь гадать. Но весьма вероятно засилье компонентов из апстрима Gnome в официальном репозитории. И прочие «бедные родственники», вроде Xubuntu и Ubuntu Mate, станут еще «бедней». А на развитии среды Cinppa в инфраструктуре Ubuntu (в частности, на Cuntu) можно ставить крест. alv@posix.ru

Сегодня мы рассматриваем:

- AMD Ryzen 7 1800X** 12
Долгожданный процессор от AMD по производительности скакнул аж на 52% относительно предыдущих. Выходит, дожидались не напрасно.
- Ryzen и Linux** 14
Как новенький процессор работает в Linux? И как насчет стратегии AMD по открытости исходного кода? Да нормально, можете не беспокоиться.
- Dell XPS 13 9360** 16
Четвертое поколение линейки Dell XPS 13 получило порты USB C, да и внутри существенно обновилось. Всё работает именно так, как полагается!
- Hitman** 17
Давненько мы не брались за оружие... от пистолета до взрывчатки, отмычек и ядов. Всякие диверсии с электричеством и газом тоже сгодятся.
- Torment: Tides of Numenera** 18
Это свободно распространяемая, насыщенная диалогами изометрическая ролевая игра, в которой думать гораздо важнее, чем действовать.
- Grim Fandango** 19
Нашего героя вдохновляет не бойня, а любовь, в эсхатологических декорациях древних индейцев. Но и без кары негодяям тут не обошлось.



► AMD Ryzen 7 1800X — процессор с фантастическим соотношением цена/качество.



► XPS13 сохраняет свою корону беспорного чемпиона на рынке ультрабуков.



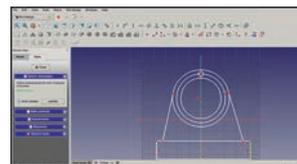
► Дизайн игры — цели, окружающая среда и задачи — силен во всех ее эпизодах.

Сравнение: Программы CAD с. 24

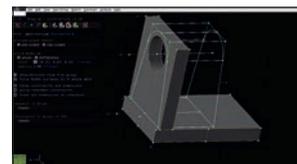
LibreCAD



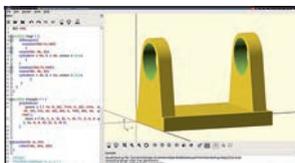
FreeCAD



Solvespace



OpenSCAD



QCAD



Зря, что ли, мы публиковали статьи про 3D-печать? Ваш новенький 3D-принтер надо кормить моделями, а модели, естественно, разрабатывать с помощью открытой программы!

AMD Ryzen 7 1800X

Зак Стори исследует процессор, который возвращает AMD в авангард доступных и тем не менее высокопроизводительных вычислений.

Спецификация

- » Разъем AM4
- » Кэш L2 4 МБ, L3 16 МБ
- » Разрядность 64 бит
- » SSE 4.1/4.2, AVX 2, AES/SHA
- » Технологический процесс 14 нм
- » Ядра 8
- » Потоки 16
- » Тактовая частота 3,6 ГГц (разблокирована)
- » Турбо 4 ГГц
- » Тепловыделение 95 Вт
- » Макс. память 64 ГБ DDR4
- » Каналы 2
- » Виртуализация AMD-V, AMD-Vi

Если вы не живете в пещере последние шесть лет, трудно не заметить учащение выпуска новых продуктов Intel и постепенный рост рынка в сегменте процессоров. Поскольку процессоры AMD серии FX вроде бы не получили признания в сфере вычислений, у Intel не было особой крайности бороться за повышение производительности своих чипов. И это сказывается. Каждое поколение процессоров Intel, начиная с Sandy Bridge и Core i5-2500K, как правило, увеличивает быстродействие IPC (команд за такт) на заурядные шесть–десять процентов. Увы, вот что получается с ненамеренной монополией.

Нельзя сказать, что у потребителей никогда не было выбора процессорных чипов. Потребителям с ограниченным бюджетом AMD поставляла более дешевые, менее мощные альтернативы, и для подавляющего большинства это было неплохим вариантом. Однако, поскольку линейки процессоров Pentium и Core i3 от Intel становились всё более энергоэффективными и мощными, спрос на 8-ядерные процессоры AMD сократился, и компания начала терять доходы. В 2016 г. одно только графическое подразделение AMD Radeon принесло больше прибыли, чем целое подразделение ЦП.

Разработка процессора требует длительного времени, и, вообще говоря, один ложный шаг способен исказить целое поколение. Вот почему в 2011 г. команда



» Генеральный директор AMD Лиза Су гордо демонстрирует первый Ryzen 7 1800X.

Красных, осознав, что они напортачили, вернулась к чертежной доске, пытаясь разработать новый чип, который заставил бы конкурентов из серии Intel Core побегать за свои денежки.

Понадобилось пять лет, чтобы разработать Zen, и еще год, чтобы его усовершенствовать и выпустить. Увеличение IPC почти на 40% на фоне флагманских процессоров AMD того времени вовсе не было легкой задачей. Но 22 февраля 2017 г. AMD, наконец, предъявила прессе потрясающий процессор линейки Ryzen, способный похвастаться впечатляющим 52% рывком по сравнению с последним поколением чипов компании.

Ядро 1800X

Новый парень на районе — Ryzen 7 1800X. Этот восьмиядерный процессор с базовой частотой 3,4 ГГц, разгоняющийся до 3,7 ГГц, и с новейшей технологией AMD XFR (экстремальная частота), способен устанавливать приоритеты и автоматически разгонять ядра под большими нагрузками до 4,1 ГГц и выше, в зависимости от температуры и способа управления напряжением. Кроме того, Ryzen использует новый SMT от AMD или одновременную многопоточность. Подобно Intel Hyper Threading, это обеспечивает каждому ядру дополнительный поток передачи данных во время вычислений и задач рендеринга, повышая производительность почти вдвое.

95-Вт тепловыделение [TDP], 16 МБ кэша L3 и 1800X дополняют — по крайней мере, на бумаге — образ ЦП-зверюги, ориентированного на профессиональных фотографов, 3D-художников и редакторов видео.

Кроме того, процессоры Ryzen основаны на 14-нм технологическом процессе с плавниковыми полевыми транзисторами [finFET], в отличие от 32-нм линейки FX, что означает лучшую удельную производительность (на ватт), меньшую потребляемую мощность и меньший размер кристалла.

Разделив каждое ядро Ryzen на составные части, AMD оснастила их 64 КБ I-кэша (используется для хранения наборов команд), 32 КБ D-кэша (используется для хранения строк данных) и 512 КБ немного более медленного кэша L2 (где хранятся команды и наборы данных). Кроме того, каждый набор из четырех ядер получает еще 8 МБ кэша L3, что в сумме составляет 20 МБ кэша на всех восьми ядрах Ryzen.

Резко контрастируя с аналогом от Intel, серия Ryzen 7 представила новый вариант взаимодействия процессора и чипсета X370 с шинами PCIe. Вообще говоря, все последние процессоры Kaby Lake поддерживают 16 линий PCIe 3.0 на самом чипе, а еще 24 линии PCIe 3.0 резервируются через чипсет Z270. Ryzen, напротив, отказался от чипсета, отдавая на него всего восемь линий PCIe 2.0, и выделив на процессор все 20 линий PCIe. Подобно своим



» Только 1700 и ниже будут поставляться в комплекте со стандартным охладителем.

Линейка Ryzen

Ryzen 5

Хотя серия Ryzen 7 от AMD формирует авангард нового штурма на переоцененную монополию Intel на процессоры, серия 5 нанесет настоящий удар своему конкуренту. Она обещает многопоточные процессоры с шестью и четырьмя ядрами, а значит, мы увидим 12-поточные маломощные чипы по цене около £200 и четырехъядерный восьмипоточный чип стоимостью £140, а то и меньше.

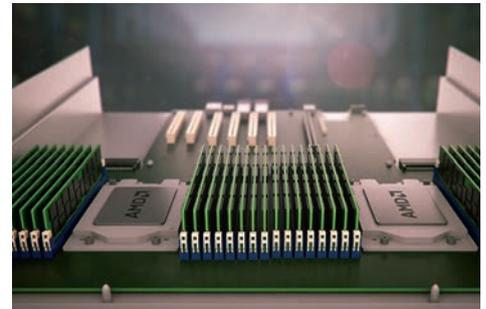
Ryzen 3

В Ryzen 3, скорее всего, мы увидим четырехъядерные чипы с более низкими тактовыми

частотами, также входящими в этот ряд. Хотя пока подробностей мало, ходят слухи, что эти чипы станут APU нового арсенала AMD, обеспечивая производительность уровня HBM и Radeon RX 460 на процессорах стоимостью менее £100.

Naples

А тут еще Naples. Серверный по замыслу компонент показывает, насколько хорошо масштабируется архитектура AMD. Чисто SOC-конструкция, 64 ядра, 128 потоков, поддержка 4 TB EEC DDR4 и 128 шин PCIe 3.0 прямо на чипе, он полностью устраняет необходимость в чипсете и топчет конкурентов.



➤ Ryzen охватит всё: от мобильных устройств до мощных серверов.

сородичам из Intel, он умеет работать с двумя графическими процессорами в Crossfire или SLI на x8, x8. Однако, помимо этого, в нем еще четыре общецелевые шины PCIe 3.0. Их можно употребить для прямого подключения к процессору либо до четырех дополнительных устройств SATA 6 Гбит/с, либо, более интригующе, одного x4 PCIe 3.0 M.2 SSD.

Больше внимания, уделяемого не материнской плате, а процессору, в плане подключаемости, отражает вековую мантру AMD, что клиенты должны иметь возможность просто обновлять чипы, а не заменять каждый раз материнскую плату. Тем не менее, чипы Ryzen поддерживаются только на собственном новейшем чипсете и разъеме AM4.

Отбросив детали, посмотрим, как он выглядит по большому счету. Ну, Intel он, считай, поднапряг. Конечно, по подключаемости чипсеты и X99, и Z270 его обгоняют, но такую производительность в пересчете

на дензнак не превзойти. Ryzen 7 1700 будет что ни день успешно бодаться с Intel Core i7-6900K. А при виде процессора, цена которого на £720 меньше, чем у команды Голубых, надо быть слабоумным либо ну очень нуждаться в лишней шине PCIe, чтобы выбрать платформу X99.

Вообще-то не всё столь безоблачно. AMD, может, и побивает Intel по производительности, но когда дело доходит до разгона, могучая линейка ядер заткнет за пояс своего краснощекого противника. При тестировании мы обнаружили, что ядра AMD в большинстве случаев способны ускориться только на 100–200 МГц по всем ядрам, в отличие от большинства разгоняемых чипов Intel, выдающих по меньшей мере 500 МГц.

Кроме того, нечто странное творится с температурой Ryzen при работе. На холстом ходу и с 280-мм жидкостным охладителем AIO эти чипы легко разогрелись до 55°C. Это немало для любого

процессора, особенно при таком эффективном охлаждении. Однако при полной нагрузке на все компоненты процессора температура поднялась всего на 13° и остановилась, практически не изменяясь. **LXF**

LINUX FORMAT Вердикт

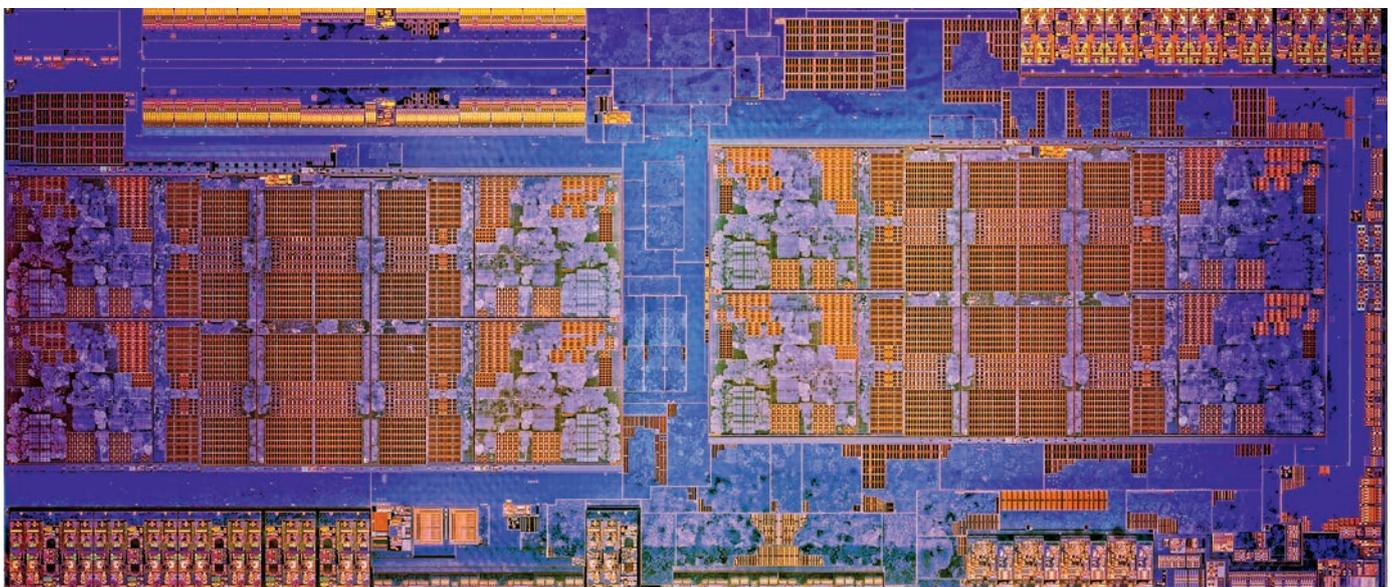
AMD Ryzen 7 1800X

Разработчик: AMD
Сайт: www.amd.com/ryzen
Цена: £490

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

» Фантастическое соотношение цена/качество выводит доступные восьмijядерные вычислительные системы в авангард массовых ПК.

Рейтинг 9/10



➤ Ryzen 7 1800X в оголенном виде.

Ryzen и Linux

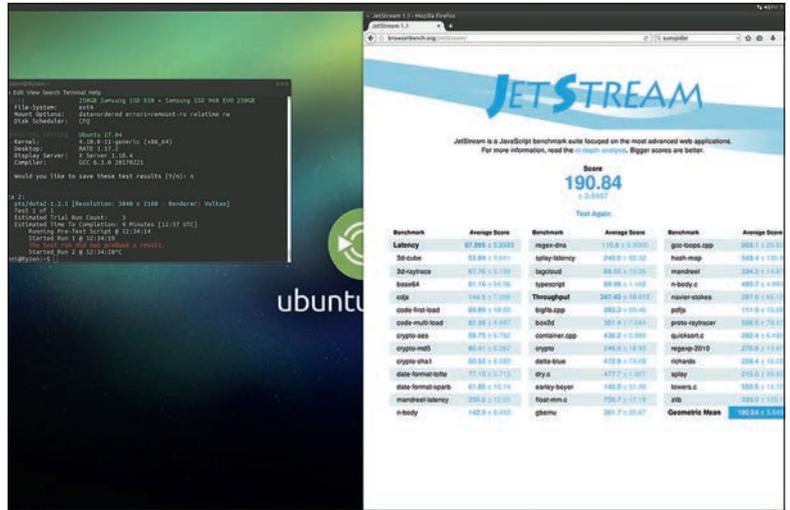
Гладко было на бумаге... но хорошо ли Ryzen поддерживается и работает в Linux? Джонни Бидвелл запускает испытательный стенд LXF.

Вкратце

» Наконец-то вышли первые процессоры AMD на новой архитектуре Zen, преемнице старой серии Piledriver. Резкое увеличение (на 52,1%) команд на такт (IPC) и XFR, повышающая частоту, дает большой потенциал. Первой на рынке появилась серия Ryzen 7 для энтузиастов. Мы протестировали самый дорогой, 8-ядерный 16-поточный 1800X.

AMD, вновь конкурирующая с Intel — это захватывающая перспектива. Свободная рыночная экономика говорит, что в итоге это должно означать лучшую производительность и посильные всем цены. По крайней мере, как только продавцы получат в ассортимент AMD. Мы видели несколько впечатляющих тестов на Windows, но видели и ряд недостатков.

Точнее, мы видели дешевый чип, исключительно подходящий для профессиональных задач типа перекодирования видео. Но также и то, что он пробуксовывает, когда дело доходит до одноядерной производительности и серьезных игр, во всяком случае, по сравнению с последним флагманом Intel Kabylake 7700K. Но как он работает в Linux? И как насчет более широкой стратегии AMD по открытию исходных кодов? Вооружившись данными обзорами, стыренными у *Maximum PC*, свежей установкой Phoronix Test Suite и ненасытной жаждой заполнения таблиц, мы даем Ryzen оценку LXF.



» JetStream измеряет производительность JavaScript, но очень сильно зависит от браузера и JS-движка. Результаты с Edge на Windows близки к 250.

В официальном объявлении о его запуске создатели стремились превознести его производительность по сравнению с топовыми чипами предыдущего поколения

Broadwell-E. Они соответствуют 8-ядерному 16-поточному составу Ryzen 7, но и стоят значительно выше бюджета многих энтузиастов игр — топовый i7 6950K продается в Великобритании-после-Brexit за £1600. Во многих отношениях 7700K является более естественным конкурентом. Конечно, у него вдвое меньше ядер и потоков, но многопоточность сложна для гетерогенных нагрузок вроде игр, так что это не будет большим недостатком. 7700K также значительно дешевле, чем Ryzen 7 1800X, участвующий в наших тестах, поэтому мы тщательно сравним эти два кремния. Сравнительное тестирование — это черная магия, и стоит помнить, что тесты Linux и Windows могут сильно отличаться. Также стоит помнить, что новое оборудование имеет болезни роста — в ближайшие недели и месяцы мы, скорее всего, услышим, что что-то работает не так, как должно, а что-то уже исправлено.

Нашей первой задачей было собрать испытательный стенд. К счастью, *Maximum PC* уже сделал это, используя вышеупомянутый топовый процессор Ryzen 1800X, 16 ГБ ОЗУ и высококлассную материнскую плату Asus AM4, ROG Crosshair VI Hero. Мы начали с установки Ubuntu 16.10, которая честно загрузилась и, похоже, работала. Однако во время тестов компиляции ядра мы столкнулись с побочными сбоями сегментации, что было странно, поскольку другие тесты работали нормально, и машина была, безусловно, стабильной. Всё прошло,

Сравнительное тестирование

Тест	AMD Ryzen 7 1800X	Intel Core i7 7700K	Разница
LAMMPS (сек.)	29,3	26,7	-9%
FFTW (Мфлп)	21 004	23 987	-14,2%
John the Ripper (взломов/сек.)	12 996	12 004	7,6%
TTSIOD (кадров/сек.)	315	227	27,7%
GraphicsMagick (ит./мин.)	242	226	-6,6%
Himeno (Мфлп)	1197	2911	-143%
Компиляция ядра (сек.)	77,4	101,5	31%
C-Ray (сек.)	8,1	19	134%
Smallpt (сек.)	38	62	63%
Stockfish (сек.)	3,600	2,905	-19,3%
Flac (сек.)	5,21	4,45	-14,6%
Lame (сек.)	9	8,5	-5,4%
FFmpeg (сек.)	13,5	7,4	-45,3%
OpenSSL (signs/сек.)	1149	839	27%
libjpeg-turbo (МПикс/сек.)	180,1	193,7	-7,5%
Blender (сек.)	566,2	518,8	-8,4%

Свобода на горизонте?

Выпуск Ryzen дал техническим специалистам много, э-э, опыта, в плане расшифровки маркетинговых выступлений, проведения независимых тестов для сравнения с обнародованными AMD, и, самое главное, решения, следует ли им инвестировать в новые чипы.

Но для приверженцев открытого ПО был еще один сюрприз: представители AMD на Reddit признали возможность того, что со временем они

выпустят необходимые биты, которые позволят свободной прошивке, такой как Libreboot, работать на компьютерах с Ryzen. Фонд свободного ПО особенно заинтересован в Platform Security Processor (PSP) AMD, а также конкурирующим Intel Management Engine. Этот код выполняется на независимых процессорах ARM, которые инициализируют ядра x86 и потенциально имеют прозрачный доступ ко всему, что система делает

потом. Но есть и другие важные составляющие, которые необходимо будет освободить, чтобы реализовать процесс свободной загрузки (см. *Вызов Libreboot AMD* на <https://libreboot.org/amd-libre>), а здесь если и будет какой-то прогресс, то, вероятно, медленный и извилистый. До 2014 г. AMD открыла исходники для своей прошивки AGESA, так что можно надеяться на возврат хотя бы к такой частичной открытости.

когда мы задействовали ядро 4.10 с kernel.ubuntu.com, но это вызвало другие проблемы — в частности, что драйвер Nvidia с ним не собирается, и мы забыли упомянуть, что в нашей машине есть еще и Nvidia 1080 [Ред.: — Почему бы LXF тоже не иметь хорошие вещи?]. Чтобы не возиться с неприятными исправлениями и ручными установками, мы совершили набег на богатый шкаф Зака Стори, стянув Radeon 470X. Поскольку AMD добавила в ядро 4.10 много кода чисто для Ryzen (часть этого кода перенесли еще и в 4.9), мы решили держаться его, но использовать вторую бету Ubuntu Mate 17.04, ради полного обновления системных пакетов. Отметим: наличие современной карты

с повторения некоторых из них, и наши результаты более или менее совпали с результатами Майкла. Ознакомьтесь с таблицей, чтобы узнать точные значения, и посмотрите сайт openbenchmarking.com, где даются детали проведения тестов.

Как подчеркивает Phoronix, по общей картине — это чип, хорошо справляющийся с нагрузками, которые эффективно масштабируются по нескольким ядрам и потокам, но затрудняющийся с нагрузками только одного ядра. Самым большим разочарованием были нагрузочный тест *Himeno Poisson* (менее чем в половину эффективный, чем на 7700K) и *Fmpeg*, где для декодирования видео h.264 потребовалось больше времени, чем чипу i5 4670, лакомому кусочку от Intel два поколения назад. Шахматы как тест вообще-то не воспринимаются всерьез, но движок Stockfish действительно обеспечивает приличную нагрузку на процессор. Он анализирует игровые деревья со многими ответвлениями (часто есть несколько ходов на выбор), развивающимися неравномерно. Так что для распараллеливания шансов много, но заданная позиция может привести к детскому мату или открыть еще много вариантов. По существу, есть множество нагрузок, разных по форме и величине, и прохождение через них всех будет проблемой для любого процессора.

Как выяснилось, древняя игра — это не какая-то странная ахиллесова пята архитектуры Ryzen: в тестах Phoronix чип несколько хуже, чем i5 4670, и мы это подтвердили. Однако проблема заключается в самом тесте, который не передает подходящие параметры *Stockfish* для многоядерного процессора и должен измерять темп теста (нод/с), а не затраченное время, хотя количество ходов (позиций) меняется с каждым прогоном. Некоторые подробности доступны в посте <http://bit.ly/2o5aFK4>. В конечном счете тест просто измеряет производительность одного ядра. Как оказалось, Ryzen — прекрасная платформа для игры в шахматы. Мы видели, что он достиг пика в 8,8 Мнод/с, используя хэш размером в 512, 12 потоков и глубину 20, что довольно бессмысленно без контекста (разные тесты

Stockfish используют разные настройки), поэтому мы протестировали его на FX8350, в свое время лидере предыдущей архитектуры Piledriver. Он достиг максимума в 5,5 Мнод/с. Интересно посмотреть, как это масштабируется с учетом количества потоков. Как можно видеть, использовать больше ядер на задачу — не всегда лучший способ решить проблему быстрее.

Ryzen — новая архитектура, и было бы безрассудно просто предположить, что все существующие бинарники поймут платформу и смогут использовать ее наилучшим образом. Замечено, что аудиокodeк ALC1220, имеющийся на нескольких платах AM4, не будет поддерживаться до ядра 4.11. Тем не менее, необходимый код будет легко перенесен, поэтому пользователям дистрибутивов с фиксированным выпуском не придется ждать слишком долго после выпуска 4.11, чтобы иметь работающий звук. Аналогично, термодатчики чипа еще не доступны в `lm_sensors`, поэтому у нас нет данных, насколько сильно разогревается чип под фланцем. Есть еще некоторые другие компоненты платформы, которые *lspci* не может идентифицировать, описывая их только как «Неосновные инструменты». Мы нашли это забавным.

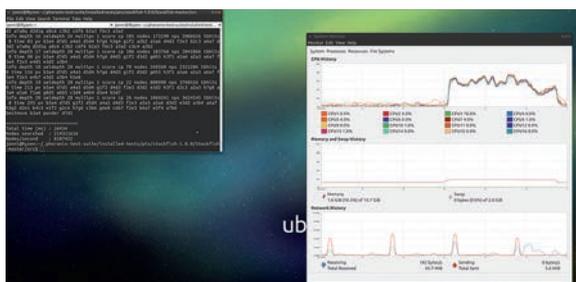
Через несколько недель после выпуска Ryzen стали появляться сообщения о том, что планировщик Windows 10 не был любезен к Ryzen, и что отключение функций Symmetric Multi-Threading (SMT) действительно улучшило производительность. Эти слухи AMD решительно отвергала, но тихой сапой внесла заплатку для ядра, подписанную в начале февраля (<http://bit.ly/2mRbh53>). Здесь мы наблюдаем, что сотрудник AMD дорабатывает код ядра, чтобы исправить топологию планирования SMT. Вероятно, они делают подобное и на Windows, но там этот процесс протекает за запертыми дверями, а с Linux всё делается в открытую.

Специфические для Ryzen оптимизации впервые появились в *GCC 6.1*, поэтому те, кто не боится компилировать собственные программы, могут попробовать переключатель `-march=zn1ver` для повышения производительности. **LXF**

Процессор, заставивший Intel струхнуть за свои денежки.

AMD означает, что мы можем воспользоваться новой моделью драйвера AMDGPU, которая позволяет иметь видеодрайверы с полностью открытым исходным кодом. Чтобы эта карта получила поддержку, которой она заслуживает, мы обновили Mesa до версии 17 (или 13.1, по старой схеме нумерации) из PPA `xorg-edgers`.

Первые результаты Ryzen 1800X для Linux на собственном тестовом комплекте опубликовал на сайте Phoronix Майкл Ларабел [Michael Larabel], и в основном они выявили исключительно функциональный процессор, безусловно заставивший Intel струхнуть за свои денежки. Мы начали



➤ Так выглядят 16 потоков, которые вместе трудятся над решением сложной шахматной задачи.

Dell XPS 13 9360

Dell обновляет свой компактный и мощный Linux ноутбук с чудесным экраном, но **Джонни Бидвелл** — неимущий писатель, и не может себе такое позволить.

Спецификация

- » ЦП 2,7 ГГц (3,5 ГГц разгон) Intel Core i7-7500U
- » Графика Intel HD 620
- » ОЗУ 16 ГБ LPDDR3 (1866 МГц)
- » Дисплей 13,3", QHD+ (3200×1800) сенсорный
- » Накопитель 512 ГБ PCIe m.2 SSD
- » Порты 1×USB 3.0, 1×USB 3.0 без питания, 1×USB-C (Thunderbolt 3), читалка SD-карт, гнездо для наушников
- » Связь Qualcomm Killer 1535 802.11ac (2,4 и 5 ГГц); Bluetooth 4.1
- » Камера 720p HD web-камера с двунаправленным цифровым микрофоном
- » Габариты 304×200×9–15 мм
- » Вес 1,29 кг

Э то четвертое поколение линейки Dell XPS 13. Хотя с виду кажется, что изменения незначительны — всего лишь порты USB C, внутри ноутбук обновлен существенно.

Главное, о чем стоит поговорить — это переход на архитектуру Kaby Lake. Это обещает увеличение производительности на 10%, увеличение времени автономной работы и, благодаря новой графике Intel HD 620, улучшенное воспроизведение видео VP9 и 10-битного HEVC. Кроме того, установлены более емкая батарея и новейший чип Wi-Fi "Killer" от Qualcomm. Заставить Linux работать с новым классным оборудованием бывает затруднительно, но Dell сделал многое, чтобы добиться этого, и хотя в некоторых тестах он отстает от Windows, всё работает именно так, как надо.

Ubuntu 14.04, поставляемая с предыдущим Developer Edition, устаревает, поэтому обновление на 16.04 весьма порадовало. Но Kaby Lake не поддается ядром ниже 4.5, а 16.04 поставляется с 4.4. Правда, установить ядро 4.8 легко, воспользовавшись недавно добавленным 16.04.2 Hardware Enablement Stack, который уже автоматически установлен в настольных компьютерах.

Поддержка Linux

Раньше пользователи Linux были в невыгодном положении, когда речь шла об обновлениях BIOS. Для обновления BIOS производители использовали исполняемые файлы DOS для загрузки в режим командной строки Windows. Теперь мы живем в мире UEFI и имеем гораздо более простой способ свободно обновлять прошивки. Это, дорогие читатели, инструмент *fwupd*, в развитие которого Dell внесла огромный вклад. Dell также предоставила несколько удобных инструментов восстановления, если дела пойдут плохо.

Когда был представлен Kaby Lake, было некоторое опасение для ноутбуков с Linux входить в глубокие C-состояния спячки, C8–C10. Однако наши тесты энергопотребления показали, что эти состояния, близкие к гибернации, достигались после относительно короткого времени простоя, а потребление энергии в режиме ожидания составляло всего 6,5 Вт. В Phoronix Test Suite (PTS), температуры по показаниям

Im_sensors иногда становились тревожными. В частности, тест *FurMark* (набравший солидные 473 балла при 1080p) показал повышение температуры ЦП до 85 градусов. Был слышен шум вентилятора, но даже при полной мощности он не раздражал. Процессор всего лишь двудерный, но всё равно собрал ядро 4.9 менее чем за четыре минуты. Тест PTS на перемножение разреженных матриц показал впечатляющие 1029,63 Мфлоп.

`dd` с параметром `conv=fdatasync,notrunc` дает скорость записи 567 МБ/с и скорость чтения (после очистки кэша) около 1,2 ГБ/с. При `trv` с выключенным радио и яркостью 50%, нам удалось получить восемь часов полноэкранный воспроизведения Sintel с 720p и пять часов с версией 4K. Оценочный тест *Unigine Heaven* при 1080p дает 17 кадров/с в оконном и 8 кадров/с в полноэкранный режимах, и это намекает, что машина не справится с играми AAA, не имея более быстрой графики Iris Pro HD.

Ноутбук может показаться дорогим, но по сравнению с аналогичными моделя-

ми Apple MacBook Pro это даром, учитывая разрешение дисплея. XPS13 сохраняет свою корону бесспорного чемпиона на рынке ультрабуков, но хотя экран QHD+ — это нечто, кое-кто предпочел бы иметь лучшее время автономной работы в версии Full HD. **LXF**

LINUX FORMAT Вердикт

Dell XPS 13 9360

Разработчик: Dell
Сайт: www.dell.co.uk
Цена: £ 1479 (£ 1180 Full HD)

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

» Как один из немногих реально поддерживающих Linux ультра-ноутбуков, Dell XPS 13 выделяется во всех отношениях.

Рейтинг **9/10**



Hitman

Устав от бесконечной реструктуризации управления, **Фил Сэвидж** отрабатывает навыки киллера в новейшем симуляторе и отводит душу всласть.

Спецификация

Минимальные:

» **ОС** Ubuntu 16.04, SteamOS 2.0

» **Процессор** Intel Core i5 2500K AMD FX-8350

» **ОЗУ** 8 ГБ

» **На диске** 68 ГБ

» **Графика** Nvidia GeForce GTX 680, AMD R9 270X, 2 ГБ VRAM

Рекомендуемые:

» **ОС** Ubuntu 16.10, SteamOS 2.0

» **Процессор** Intel Core i7-3770

» **ОЗУ** 16 ГБ

» **На диске** 68 ГБ

» **Графика** Nvidia GeForce GTX 970, AMD R9 290, 4 ГБ VRAM

NB: нет поддержки графических карт Intel, драйвера Nvidia 375.26, AMD Mesa 13.0.3

Нас, как и многих, удивила новость о том, что *Hitman* будет выходить по частям. Удивила, но не обеспокоила. Ведь в этой серии сюжет не главное. Выпустив миссии по отдельности, IO, казалось, сигнализировало о возвращении *Hitman* в качестве отдельной серии игр в жанре учебного убийства. Всё, что так любил поклонники *Blood Money*, только больше и разнообразнее.

Зачастую можно просто стрелять в свою цель и бежать, надеясь добраться до выхода, прежде чем вас окружит команда секьюрити. Это действия правильные, но бесполезные. Агент 47 — профессионал, а значит, вы должны появляться и исчезать практически бесшумно. Самая трудная, но и приносящая больше всего удовлетворения задача в *Hitman* — это устранить цель так, чтобы вообще себя не выдать. Вот тут-то в ход идет переодевание.

Переодевание имело в этой игре ключевое значение с момента ее создания, но сегодня *Hitman* предлагает самую элегантную реализацию этой функции: здесь вы можете быть обнаружены только определенными NPC того же типа. Иногда это вызывает недоверие — не всегда понятно, почему одни охранники могут распознавать своих коллег, а многие другие не могут; но прозрачность тоже приветствуется, она дает гарантию, что вы сможете обойти этих персонажей, оставаясь незамеченным.

Достижение цели требует не только использования маскировки. Вам также следует придумать, как именно вы собираетесь это сделать. У вас, как всегда, есть пистолет с глушителем, прекрасно для этого подойдет. Или же вы можете в каждой миссии пользоваться разными инструментами, от оружия и взрывчатки, до отмычек



» **Снайперская винтовка разрешена для устранения объекта, но куда ей до лося...**

и различных ядов. Нередко, однако, самые коварные средства для убийства подсказывает логика уровня. 47-й умеет перепутывать электрические кабели, ронять люстры или портить газовые обогреватели. Помимо более общих вариантов, в каждой миссии доступны свои собственные методы убийства: «Opportunities [Возможности]».

Возможности позволяют вам манипулировать привычками своих жертв, часто фатально для них. Как и аварии в *Blood Money*, это мощный инструмент для того, чтобы либо отвести цель от секьюрити, либо привести ее прямо к гибели. По умолчанию, Возможности приводят вас, шаг за шагом, именно туда, куда вам надо. Если для того, чтобы какого-нибудь негодяя придавило чучелом лося, нужно переодеться кинооператором, вам подскажут, как это сделать. К счастью, любой элемент интерфейса пользователя перенастраивается. Возможности можно задать минимальные, только с самыми общими наводками, и самим продумывать более тонкие детали. Результат будет тот же — смерть негодяя от лося; но чувство выполненного долга будет куда больше. IO удалось сделать уровни более длинными, чем в любом из предыдущих *Hitman*'ов, и очень насыщенными — и атрибутами, и возможностями. Но столь содержательных, как в *Blood Money*, мало: ничего такого культового, как Марди Гра [Mardi Gras] в Новом Орлеане, нет, но в каждом есть что-то свое. Каждый уровень предусматривает повторы прохождения: либо выполняйте ту же миссию другими методами, либо выберите один из вариантов Escalations [Эскалаций] — специаль-

ных миссий с другими целями, усложняющихся с каждым новым прохождением. Таким образом, каждая игровая среда оказывается достаточно объемной и сложной, чтобы к ней можно было обратиться несколько раз.

Если убрать оформление и импровизацию, вы увидите внутренний механизм — с виду идеальную систему циклов, которые взаимно пересекаются, создавая сложную сцену. Вам остается решить, как вы хотите ее разрушить: хирургически удалив ключевых актеров или разгромив всё это оружием, бомбами и чучелом лося.

Сам по себе дизайн — цели, окружающая среда и задачи — силен во всех шести эпизодах. *Hitman* не совершенен, но качественно сделан, интересен, в нем нет таких ошибок, как в *Absolution*, и, кроме того, это возвращение к форме игры-сериала. **LXF**



LINUX FORMAT Вердикт

Hitman

Разработчик: Feral Interactive

Сайт: www.feralinteractive.com

Цена: £40 (Эпизод 1 — £11)

Сюжет	8/10
Графика	8/10
Увлекательность	9/10
Оправданность цены	7/10

» Уверенный дизайн и тяга к эксперименту позволяют создать лучшие миссии за всю долгую карьеру агента 47.

Рейтинг 8/10

Torment: Tides of Numenera

Земля в руинах, повсюду обломки старых технологий, средневековые порядки. Крис Терстен бросил дела, чтобы поиграть в наследника Planescape, *Torment*.

Спецификация

Минимальные:

- » ОС Ubuntu 16.04 (64-бит), SDL 2.0
- » Процессор Intel Core i3
- » ОЗУ 4 ГБ
- » Графика NVIDIA GeForce GTX 460 или подобный
- » На диске 15 ГБ

Рекомендуемые:

- » ОС Ubuntu 16.10 (64-бит), SDL 2.0.5
- » Процессор Intel i5
- » ОЗУ 8 ГБ
- » Графика NVIDIA GeForce GTX 680 или подобный (драйвер 375.26)

Torment: Tides of Numenera обещает стать преемником игры *Torment*, ставшей некогда родоначальником жанра. Это насыщенная диалогами изометрическая ролевая игра с путешествием, в котором думать гораздо важнее, чем действовать. Сражаться тоже надо, но пройти кампанию удастся и без насильственных действий.

Знакомая обстановка Planescape ушла, на смену ей пришел Девятый мир [Ninth World] — там, в далеком будущем, разворачивается действие нарисованной на бумаге игры Numenera от Монте Кука [Monte Cook]. Это Земля через миллиард лет, технологический осколок бесчисленных исчезнувших цивилизаций, ставший местом обитания средневекового общества. Там есть магия, но на самом деле это наука, и наука странная, где возможны путешествия во времени, перенос сознания, параллельные вселенные и наномашины.

Вы — Последний Изгой [Last Castoff], бессмертное творение Изменчивого Бога [The Changing God], футуристического волшебника, который переносит свое сознание в новые тела, искусственно продлевая себе жизнь. Изгоев, подобных вам, преследует таинственный убийца по имени Скорбь [Sorrow], и желание избежать, казалось бы, неминуемой участи мотивирует вас сделать первые шаги в Девятый мир. Ваши решения тоже очень значимы: сделав в самом начале игры шаг, за которым, как мы ожидали, последует надпись «Игра окончена», мы существенно изменили начало квеста.



» Девятый мир представляет собой странную смесь технологий.

Выбирая, скажем, такую черту, как чтение мыслей, вы можете быть уверены, что вся кампания тщательно продумана с учетом этого.

В первые часы игры ваш темп замедлен. *Numenera* приходится знакомить вас с Девятым миром, а также со своей собственной сложной его интерпретацией; для нового игрока это означает много незнакомой терминологии. Но зато к тому моменту, когда сюжет действительно начнет развиваться, у вас будут все необходимые знания.

Если вы любите поразмышлять о статистике элементов и стратегии своей партии, то это не ваша игра. Здесь то, как вы будете использовать возможности, скорость и умственные способности для достижения целей вашего персонажа, определяет довольно простая система умений — начиная от того, как взобраться на стену, и заканчивая тем, как убедить кого-то разделить вашу точку зрения.

Впечатляют в этой системе ее способности адаптации к разнообразным обстоятельствам. Выполнение задачи происходит одинаково, делаете ли вы это посредством диалога в открытом мире, работая через Омут [Merge] (фрагмент памяти, который принимает форму иллюстрированного квеста, основанного на выборе), или борьбы. Во время опасных столкновений игра становится пошаговой, и могут применяться специальные боевые умения, но ваш обычный набор навыков всегда с вами, и если

вы решить исход боя, поговорив, спрятавшись или вернув к жизни какое-нибудь древнее устройство, такая возможность у вас есть. Сценарии вызывают довольно сильные эмоции, но их художественное воплощение — скорее смешанные. Местами впечатление очень цельное, но кое-что можно было бы и переделать. Ваши спутники иногда перемещаются хаотично и застревают в пейзажах, и хотя это не мешает вам двигаться, но всё же странно. Анимация тоже бывает неуклюжей.

Torment — странный зверь: игра для книжечеев и квест для тех, кто не особо хочет драться. Но здорово, что она опять у нас есть. **LXF**

LINUX FORMAT Вердикт

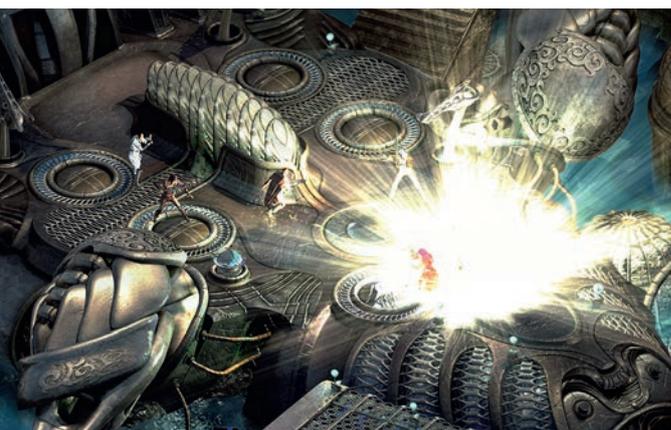
Torment: Tides of Numenera

Разработчик: inXile Entertainment
Сайт: <http://bit.ly/1tbGxb2>
Цена: € 35

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

» Слегка затянутый в начале, но дающий пищу для ума квест в потрясающей обстановке. Достойное продолжение любимой ролевой игры.

Рейтинг **7/10**



Grim Fandango

Евгений Балдин вынужден признать, что ему нравятся детективы и «истории любви»; правда, они должны быть качественными.

Вкратце

» Классика квеста еще от LucasArts. Живая легенда Тима Шейфера (Tim Schafer), на продлевающих жизнь стероидах.

Если вам хоть немного нравятся истории в стиле «нуар», то *Grim Fandango* пропускать нельзя. В него просто необходимо погрузиться. Если нет желания решать игровые загадки этого мира, можно воспользоваться многочисленными пошаговыми руководствами. Даже в этом случае удовольствие от просмотра красивой истории совершенно затмит и не самое удобное управление, и устаревшую на 19 лет графику. Не удастся задействовать обновленную графику — не страшно: основной графический ускоритель находится в вашей голове, и он точно не подведет. Головоломки в игре подчас весьма интересны и осмысленны. Например, чтобы перегрузить электрическую сеть, главный герой Мэнни должен открыть морозилку холодильника, что понижает его КПД и, соответственно, вышибает пробки. Мелочь, конечно, физическая, но почему-то редко встречающаяся мелочь.

Основным прародителем *Grim Fandango* является широко известный Тим Шейфер — известный еще и неоднозначностью своих проектов. Так, восторги по поводу 1-й части квеста *Broken Age* по накалу могут сравниться только с гневными отзывами о безобразно исполненной завершающей 2-й части. И всё равно нельзя не признать, что *Grim Fandango*, несмотря на ряд весьма спорных и экспериментальных для своего времени решений, является шедевром.

Мир *Grim Fandango* весьма необычен. Тим Шейфер специально изучал индейские легенды о мире смерти, чтобы использовать эту фактуру в качестве декораций. Эта этническая смесь разбавлена стилистикой голливудских криминальных драм 1940-х



» Разработчики снабдили своего героя навыками хакера. В игре это ему не особо пригодилось, но выглядит весьма современно.

годов. Здесь почти все курят; правда, следует отметить, что все курящие уже мертвы. Сюжет вращается вокруг необходимости каждому приличному умершему добраться до Девятого загробного мира. Праведникам полагается билет на экспресс, а обычные люди идут пешком. Махинации на рынке предоставления транспортных услуг умершим и несправедливость по отношению к истинной праведнице Мерседес Коломар сподвигли туристического агента Мануэля Калаверу стать героем и отправиться в путешествие длиной в четыре года. Мотивацией для Мэнни явно было нечто большее, чем желание исправить ошибку.

Grim Fandango был первым квестом LucasArts, применившим 3D-модели. Для этого на смену популярному старичку SCUMM был разработан принципиально новый игровой квестовый движок GrimE. Побочным следствием этого решения является популяризация языка Lua в игровой индустрии. Он был встроен в GrimE. В порядке инновации было решено отказаться от управления “point-and-click”, которое, к счастью, вернулось в расширенном (Remastered) издании. Возможность управлять героем по-прежнему, то есть с клавиатуры, тоже сохранена, так что те, кто встречался с игрой до переиздания, могут предаться ностальгии. Остальным же лучше остаться с мышью.

К вопросу об обновленной графике переиздания. К сожалению, далеко не всегда

она доступна, и запуск на ноутбуке чреват черным экраном. Чернота убирается выбором оригинальной отрисовки. Сделать это можно, выйдя в настройки по клавише Esc, выбрав раздел Special Features и переключившись в Original rendering вместо Remastered. Тип отрисовки переключается на лету клавишей Backspace. Официально русский язык не поддерживается вообще. Для версии Remastered есть любительский перевод, причем утверждается, что Linux-порт тоже поддерживается. Как и в самой игре, тут не всё гладко, но все эти шероховатости не портят удовольствия от погружения в мир смерти, в котором жизнь-таки продолжается. **LXF**

Свойства навскидку



Ваша цель...

У мира существует край, и вам волей-неволей придется там побывать, чтобы наказать негодяев и спасти невинных.



Награда герою

Классические нуар-истории просто обязаны иметь хороший конец. Это не закон жанра, но приятное к нему дополнение.

LINUX FORMAT Вердикт

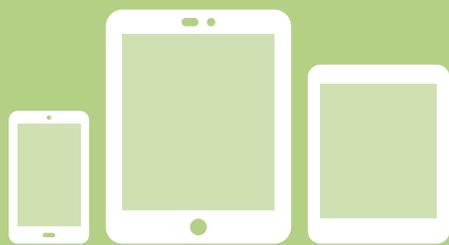
Grim Fandango

Разработчик: LucasArts Entertainment, Double Fine Productions (переиздание)
Сайт: www.grimremastered.com
Цена: 349 руб. в Steam или gog

Сюжет	10/10
Графика	5/10
Увлекательность	8/10
Оправданность цены	9/10

» Детективная история любви в декорациях ацтекских мифов о загробной жизни.

Рейтинг 8/10



ХРОМБУКИ

Новый трансформер от Lenovo

Поддержка Android-приложений позволит хромбукам обойтись без Интернета.

Lenovo Flex 11 — типичный представитель нового поколения поддерживающих приложения для Android хромбуков, которые выходят на рынок в этом году. Его начальная цена — \$279, а главная особенность — способный вращаться на 360° вокруг своей оси сенсорный дисплей, что позволяет превращать Flex 11 из «раскладушки» в планшет или фиксировать в любом из промежуточных положений.

Ранее определенная защита была прикреплена в основном хромбукам, ориентированным на студентов, и лишь единичным моделям общего назначения (например, Asus C202S). Ударопрочная конструкция Flex 11 позволяет выдерживать падения с высоты 2,4 фута (около 75 см), все его порты и тачпад сделаны водонепроницаемыми;

клавиатура оснащена специальными каналами для стока случайно пролитой на нее воды (до 330 мл), а корпус имеет нескользящую текстуру для надежного хвата. Операционная система — Chrome OS, однако доступ к Google Play с возможностью установки приложений для Android позволяет новым хромбукам (на Западе их называют Android-app-loving) работать и при отсутствии Интернета и доступа к облачным сервисам Google.

Спецификации Flex 11 достаточно типичны: 11-дюймовый сенсорный дисплей 1366×768 с поддержкой до 10 одновременных касаний и антибликовым покрытием; 4-ядерный CPU ARM 2,1 ГГц; 4 ГБ ОЗУ LPDDR3; порт USB 3.1 Type-C для передачи данных и зарядки, плюс традиционный порт USB-A; Wi-Fi 802.11 a/g/n/ac; порт

► **Lenovo Flex 11:** недорогой компактный компьютер для несложных задач и просмотра видео.



HDMI, 32 ГБ ПЗУ eMMC и слот для карт SD; веб-камера 720p над дисплеем; размеры 296×206×21,2 мм; вес 1,35 кг; аккумулятор рассчитан на 10 часов автономной работы. Стилуса у Lenovo Flex 11 нет, что, впрочем, не удивительно: только новые хромбуки Samsung предлагают тонкий стилус в подпружиненном слоте.

НОВИНКИ BLACKBERRY

Энергоэффективный телефон

BlackBerry и TCL начали поставки Android-смартфона с аппаратной клавиатурой.

В конце февраля на Mobile World Congress в Барселоне китайская компания TCL Communication, получившая лицензию на долгосрочное производство Android-устройств под брендом BlackBerry, показала новый смартфон BlackBerry KEYone. Продажи должны были начаться в апреле по цене от \$549, но в конце апреля TCL объявила: продажи KEYone в США и Канаде стартуют 31 мая, причем, по прогнозам аналитиков, задержка им не повредит — отчасти потому, что в мире около 275 млн пользователей BlackBerry, многим из которых необходим новый смартфон именно с аппаратной клавиатурой. Подобные клавиатуры многие годы остаются хитом в таких странах, как Индия и Индонезия — основных потребителей продукции BlackBerry и BlackBerry Messenger (в начале мая Роскомнадзор заблокировал в РФ мессенджеры BlackBerry

Messenger, Imo и Line, а также чат Vchat, так как все эти ресурсы не желали войти в реестр организаторов распространения информации). Аналитик J. Gold Associates Джек Голд [Jack Gold] не считает большой проблемой задержку релиза на 4–5 недель. Того же мнения аналитик IDC Брайан Бассетт [Bryan Bassett]: «Своевременный выпуск устройства с проблемами повредил бы бренду BlackBerry намного сильнее, чем выпуск устройства позже обещанной даты».

KEYone комплектуется процессором Snapdragon 625, 3 ГБ ОЗУ и 32 ГБ встроенной памяти (поддерживаются карты microSD объемом до 2 Тб), основной камерой Sony IMX378 разрешением 12 Мпикс, поддерживающей запись видео 4К, фронтальной 8 Мпикс и 4,5-дюймовым сенсорным экраном с разрешением 1620×1080 и плотностью 433 PPI; под экраном размещена аппаратная четырехрядная QWERTY-

► **В KeyOne классический форм-фактор BlackBerry сочетается с полной экосистемой Android и безграничными возможностями выбора приложений.**



клавиатура с традиционной для BlackBerry поддержкой «быстрых» клавиш (настраиваемых на определенный тип нажатия ярлычков: например, длительное нажатие на клавишу F запускает Facebook); доступны и привычные пользователям ПК сочетания Ctrl+. Аккумулятор емкостью 3505 мА·ч поддерживает быструю зарядку QC3.0, обеспечивающую уровень заряда 50% за 36 мин. Размеры KEYone — 149,3 мм×72,5 мм×9,4 мм, вес — 180 г.

АВТОМОБИЛЯМ

Mali-C71: глаз — алмаз!

Новый процессор изображений от ARM улучшит «цифровое зрение» автомобиля.

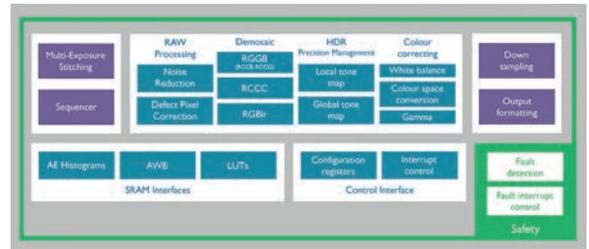
Автомобили стремительно превращаются в компьютеры с уникальным набором требований: если на ПК камера с сопутствующей функциональностью вторична, то в автомобиле уже сейчас она является одним из наиболее важных компонентов (системы самопарковки); а в автономных автомобилях камеры буквально станут «глазами», помогая захватывать и анализировать изображения.

Компания ARM представила Mali-C71 — специализированный процессор обработки сигналов, поступающих с камер автомобиля, среди возможностей которого: анализ полученных изображений и идентификация людей — как стоящих, так и, например, переходящих дорогу — а также дорожных знаков и полос движения, в различных условиях освещения и видимости; определение погодных условий (по информации, полученной от GPS); идентификация (с помощью установленной в салоне камеры) засыпающего водителя с выдачей предупреждения. Аналогичные функции могут

выполнять ориентированные на автономные транспортные средства графические процессоры таких компаний, как Nvidia, однако решение ARM является значительно более энергоэффективным.

В ближайшие годы в каждом автомобиле будет установлено более 10 камер, и зависимость автономного автомобиля от них будет только возрастать. Каждый Mali-C71 поддерживает до четырех камер в режиме реального времени, система может состоять из нескольких таких чипов. Mali-C71 предназначен, в первую очередь, для автомобилей, управляемых водителем, однако имеет функциональность, которую можно будет использовать в полностью автономных автомобилях. Поддерживаются изображения с разрешением 4096×4096 пикс, обеспечиваются 24 «стопа» динамического диапазона.

Процессоры сигналов изображения — категория не новая; уже сегодня существуют их мобильные реализации. Отличительной чертой Mali-C71 является сверхвысокая надежность, гарантирующая достоверное



► Структурная схема Mali-C71, первого продукта в серии ISP Mali Camera от ARM Imaging & Vision Group.

распознавание пикселей и отсутствие ошибок в данных. Согласно заявлению ARM, функциональность, качество изображения и элементы безопасности, которыми обладает Mali-C71, допускают использование новинки как в многокамерных системах помощи при парковке, так и в полностью автономных транспортных средствах.

Чип может работать не только с ARM, но и с другими архитектурами (да, борьба между архитектурами ARM, x86, Power и MIPS ведется, в том числе, и за место в автомобиле!).

ТРАНСФОРМЕРЫ

Где тонко, там и... лучше

Новый гибридный планшет HP Elite x2 1012 G2 обгоняет Microsoft Surface Pro 4.

Толщина корпуса обновленного HP Elite — 0,36" (9,1 мм), а комплектуется он процессорами Intel Kabylake, которых нет в последней версии Surface Pro — в момент ее выхода такие процессоры Intel еще не поставляла. Покупателям Surface Pro 4 предлагается один из CPU Core 6-го поколения (или Skylake), в то время как x2 1012 G2 может быть оснащен Core i3-7100U, Core i5-7200U или i5-7300U, или же Core i7-7600U, при этом ни один из этих CPU не использует интегрированную графику Intel Iris, которая включена в чипы Surface Pro i7. Как и планшет 2-в-1 от Microsoft, x2 1012 G2 поддерживает до 16 ГБ RAM и твердотельный накопитель (SSD) емкостью до 1 ТБ. HP увеличила разрешение и размер оснащенного защитным стеклом Corning Gorilla Glass 4 сенсорного дисплея: 2736×1824 пикс и 12,3" против 1920×1080 пикс и 11,6" у оригиналь-

ного Elite x2 1012 G1, теперь эти параметры сравнялись с Surface Pro. Планшет оснащен портом USB Type-C с поддержкой Thunderbolt 3 и быстрой зарядки HP Fast Charge, предусмотрены ИК-камера и сканер отпечатков пальцев.

По сравнению с последней версией Surface Pro, гибридный HP имеет еще несколько преимуществ. Он выполнен в соответствии с требованиями MIL-SPEC (система стандартов министерства обороны США), комплектуется подключаемой клавиатурой HP Collaboration Keyboard и цифровым пером Wacom. Компания упростила процесс вскрытия устройства для проведения ремонта или модернизации (известная проблема аппаратных средств Surface Pro).

А вот цена у Surface Pro 4 привлекательнее. Самый дешевый планшет Surface можно приобрести за \$799, тогда как базовая версия Elite x2 обойдется на \$300 дороже.

Но при этом в базовой версии Surface Pro 4 пользователь не получит клавиатуру или перо, и ему придется довольствоваться процессором Intel Core m3 вместо более производительного Core i3. Впрочем, потенциальным покупателям нового Elite x2 1012 G2 придется немного подождать: HP обещает выпустить его в июле. **LXF**



► Новое поколение гибридных планшетов Elite x2 1012 G2 отличается увеличенным дисплеем, улучшенной аппаратной платформой, наличием подключаемой клавиатуры и пера Wacom.

Tizen: Свое, родное

Лада Шерышова принимается за создание простого мобильного приложения на Tizen.



Наш эксперт

Лада Шерышова долгие годы работала на коммерческие корпорации, создавая промышленные высоконадежные информационные системы. Но пришло время сбросить оковы и применить свои знания и опыт в работе со свободным ПО.

Мобильное нативное («родное») приложение для ОС Tizen создается на языке C и может быть запущено на устройствах, поддерживающих платформу Tizen. Такое приложение использует нативный API, который предоставляет различные интерфейсы к аппаратным возможностям устройства.

Создадим и запустим простое нативное приложение, которое использует готовый шаблон и отображает на экране мобильного устройства текст «Hello Tizen». При этом никакого взаимодействия с пользователем происходить не будет. При нажатии на клавишу Back приложение перейдет в фоновый режим.

Для разработки приложения нам понадобится инструмент Tizen IDE. Это интегрированная среда разработки, входящая в состав Tizen SDK. Она основана на Eclipse и содержит набор необходимых инструментов для разработки приложений под Tizen. Tizen IDE включает перспективы Tizen Native и Tizen Web для разработки нативных и web-приложений соответственно.

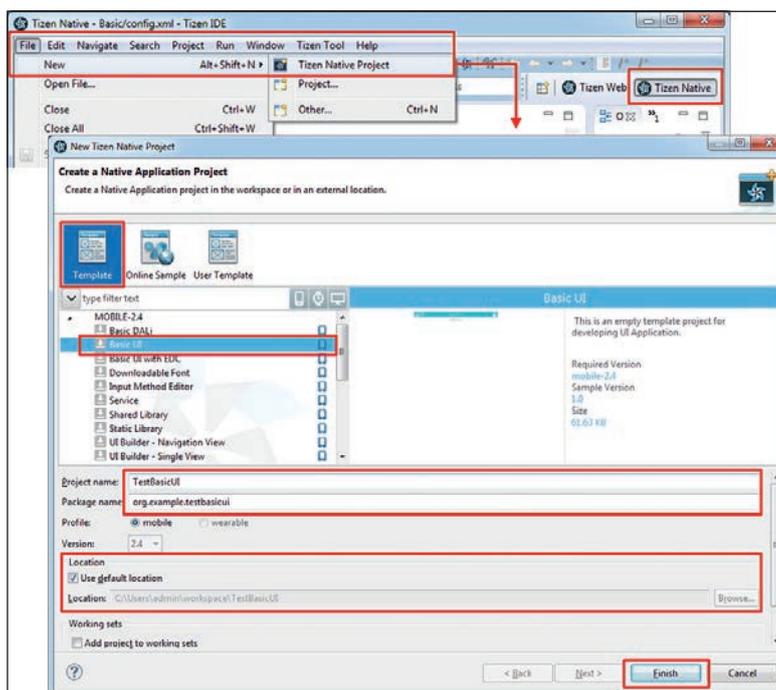
Приступим к разработке. Если у вас еще не установлен Tizen SDK, то необходимо скачать и установить его на своей платформе. Мы не будем подробно останавливаться на установке Tizen SDK, поскольку процесс инсталляции достаточно прост и не вызывает никаких вопросов. Процесс разработки нативного приложения состоит из следующих шагов:

- 1 Создание нативного проекта для мобильного устройства.
- 2 Конфигурация приложения.
- 3 Реализация функциональности приложения.
- 4 Сборка и компиляция приложения.
- 5 Генерация сертификата.
- 6 Запуск приложения.

Шаг 1. Создаем проект нативного приложения

- 1 Запустите Tizen IDE.

➤ Рис. 1. Создание нативного проекта Tizen.



2 В верхнем правом углу выберите перспективу Tizen Native. Если перспективы не отображаются, в главном окне Tizen IDE выберите Window > Open Perspective > Other... > Tizen Native и нажмите Ok.

3 Создайте новый проект одним из следующих способов:

» В главном меню Tizen IDE выберите File > New > Tizen Native Project или File > New > Project... > Tizen Native Project.

» В Project Explorer щелкните правой кнопкой мыши и выберите New > Tizen Native Project или New > Project... > Tizen Native Project. Запустится мастер создания проектов Project Wizard.

4 В Project Wizard укажите детальную информацию о проекте (рис. 1). Мастер создания проектов используется для создания каркаса приложения с необходимой структурой папок и обязательных файлов. Project Wizard позволяет легко создавать различные приложения путем выбора соответствующего шаблона или образца. Для нашего примера выберите тип проекта Template. На панели слева выберите профайл и версию платформы (MOBILE 2.4), раскройте дерево и выберите тип нативного приложения — BasicUI. В поле Project Name введите наименование проекта (от 3 до 50 символов), а в поле Package Name — наименование пакета. Их можно оставить как есть по умолчанию. Главное, чтобы они были уникальны, иначе Tizen IDE выдаст ошибку. В разделе Location оставьте отмеченной опцию Use default location, если хотите использовать местоположение файлов проекта по умолчанию, либо снимите галочку и в поле Location выберите другой путь к файлам проекта. Нажмите кнопку Finish. Project Wizard начнет создание файлов проекта приложения на основе контента выбранного шаблона. Этот процесс требует времени. Поэтому процесс создания проекта приложения можно запустить в фоновом режиме, нажав кнопку Run in Background.

Перейдите в Project Explorer и раскройте дерево нового проекта. Здесь вы увидите созданные папки и файлы вашего проекта (рис. 2). К наиболее важным папкам относятся:

- » inc Папка по умолчанию для включенных в проект исходных файлов.
- » res Папка для файлов ресурсов, используемых только вашим приложением.
- » shared Папка для файлов ресурсов, доступных другим приложениям.
- » src Папка для файлов с исходным кодом.
- » lib Папка для файлов внешних библиотек.
- » tizen-manifest.xml Файл декларации, используемый платформой для установки и запуска приложения.

Шаг 2. Конфигурируем приложение

Вы можете просматривать и редактировать конфигурацию приложения в редакторе Tizen Manifest Editor. Для этого в Project Explorer дважды щелкните по файлу **tizen-manifest.xml** в вашем проекте. На панели справа откроется вкладка редактора, где вы можете выполнить все необходимые настройки. Для нашего приложения мы конфигурацию изменять не будем, а просто рассмотрим возможные настройки приложения. Эти настройки сгруппированы по следующим вкладкам (рис. 2):

- » Overview Задаёт основную информацию, такую как пакет, метка, версия, иконка, автор приложения и т.д.
- » Features Задаёт требуемую функциональность программного и аппаратного обеспечения. Эта информация используется для фильтра приложений в магазине Tizen.

- » Privileges Задаёт привилегии API или группы API, к которым имеет доступ и которые использует приложение.
- » Localization Определяет локализованные значения метки, описания и иконки приложения для выбранных языков (локалей).
- » Advanced Задаёт расширенные возможности, такие как метаданные приложения (определённые пользователем пары ключ-значение), управление данными для сервисов, управление функционалом приложения и т.д.
- » XML Source Просмотр и редактирование кода исходного файла декларации **tizen-manifest.xml**. Изменения, выполненные и сохранённые на других вкладках, автоматически отображаются в коде исходного файла и наоборот. Эта вкладка предназначена только для продвинутых пользователей.

Для сохранения изменений в меню Tizen IDE выберите File > Save All (или нажмите комбинацию клавиш Ctrl+S).

Шаг 3. Реализуем функциональность

После того, как вы создали проект и сконфигурировали приложение, можно приступать к реализации требуемой функциональности приложения. В нашем примере мы будем использовать только стандартные функции из шаблона проекта, поэтому никакого изменения кода не потребуется. Переходим к следующему шагу.

Шаг 4. Выполняем сборку приложения

Процесс сборки приложения включает проверку и компиляцию файлов проекта. Приложение может быть построено следующими способами:

» **Автоматически** Это означает, что Tizen IDE автоматически перестраивает приложение, когда вы вносите изменения в файл источника или файл ресурса и сохраняете проект приложения. Для включения автоматической сборки приложения:

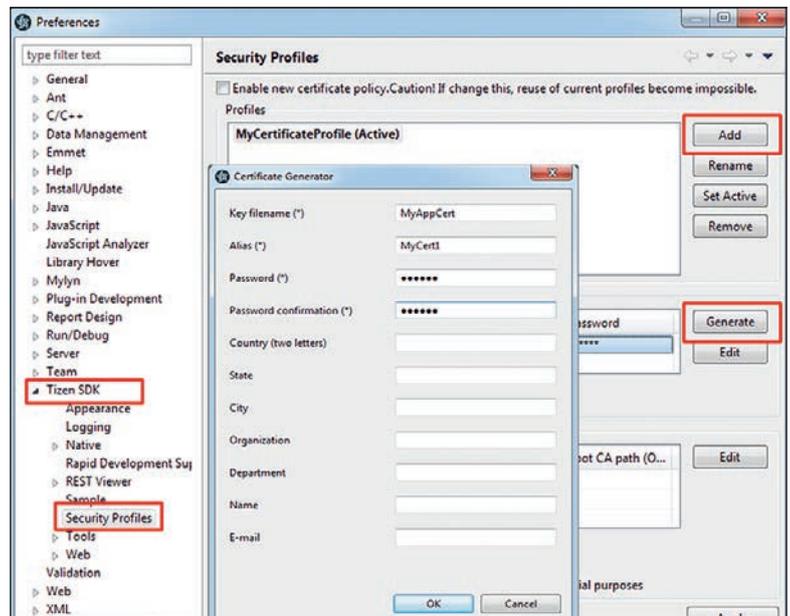
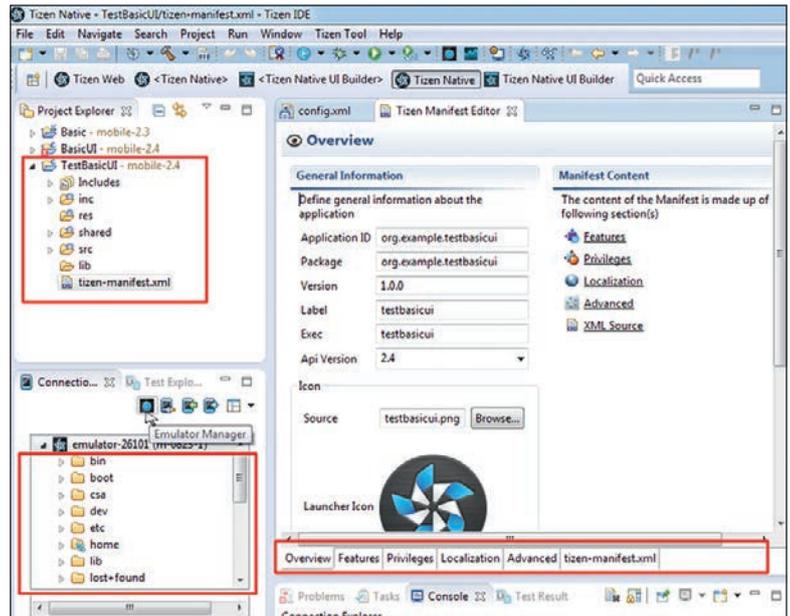
- 1 Выберите проект в Project Explorer.
- 2 В меню Tizen IDE выберите Project > Build Automatically. Рядом с этой опцией меню появится галочка. Для отключения автоматического построения приложения ещё раз выберите Project > Build Automatically.

» **Вручную** Этот режим позволяет вам самим определять, когда построить приложение. Чтобы построить приложение вручную, в Project Explorer щёлкните правой кнопкой мыши по проекту и из контекстного меню выберите Build Project, либо в меню Tizen IDE выберите опцию Project > Build Project, либо нажмите клавишу F10. Существует 2 варианта конфигурации сборки приложения. Чтобы просмотреть активную конфигурацию или изменить её, в Project Explorer щёлкните правой кнопкой мыши по проекту и выберите Build Configuration > Set Active, и далее — 1 Debug или 2 Release. По умолчанию включена опция Debug. Она предназначена для отладки и тестирования приложения. Для этого режима скомпилированные двоичные и отладочные файлы будут храниться в папке **Debug** в проекте приложения. Опция Release используется для тестирования готовой к выпуску версии приложения. В этом случае скомпилированные двоичные файлы будут храниться в проекте в папке **Release**.

Выполните сборку приложения одним из описанных выше способов. Если при сборке приложения не возникло ошибок, папка проекта **Project Explorer** будет отмечена значком "C".

Шаг 5. Генерируем сертификат

Любое приложение, созданное для Tizen, должно быть подписано сертификатом разработчика. Он гарантирует, что код приложения не был изменён с момента его публикации. Для создания своего сертификата в меню Tizen IDE выберите Window > Preferences. В открывшемся окне выберите Tizen SDK > Security Profile. В разделе Profiles создайте новый профайл. Для этого нажмите кнопку Add и введите наименование профайла. Далее в разделе Author Certificate нажмите кнопку Generate и заполните требуемые поля (рис. 3).



На целевом устройстве надо проверить дату: если она меньше, чем время генерации сертификата, то устройство будет отвергать сертификат. Теперь можно запускать приложение.

Шаг 6. Запускаем приложение

Запустить приложение можно либо в эмуляторе, либо на реальном устройстве. Запустим приложение в эмуляторе. Для этого:

1 В Tizen IDE запустите Emulator Manager с панели Connection Explorer, кликнув по соответствующей иконке, либо выберите пункт меню Tizen Tool > Emulator Manager. В Emulator Manager выберите из списка эмулятор мобильного устройства и щёлкните по иконке launch emulator. Если в списке нет нужного эмулятора, создайте новый. Когда эмулятор мобильного устройства запустится, вы сможете увидеть его на панели Connection Explorer. Он представлен как древовидная структура папок; просмотреть её можно, нажав на стрелку слева от наименования экземпляра эмулятора (рис. 2).

2 Запустите приложение. Для этого в Project Explorer кликните правой кнопкой мыши по проекту и выберите Run As > Tizen Native Application. Также можно запустить приложение из главного меню Run, либо нажать комбинацию клавиш Ctrl+F11. Ваше приложение автоматически загрузится в эмулятор! LXF

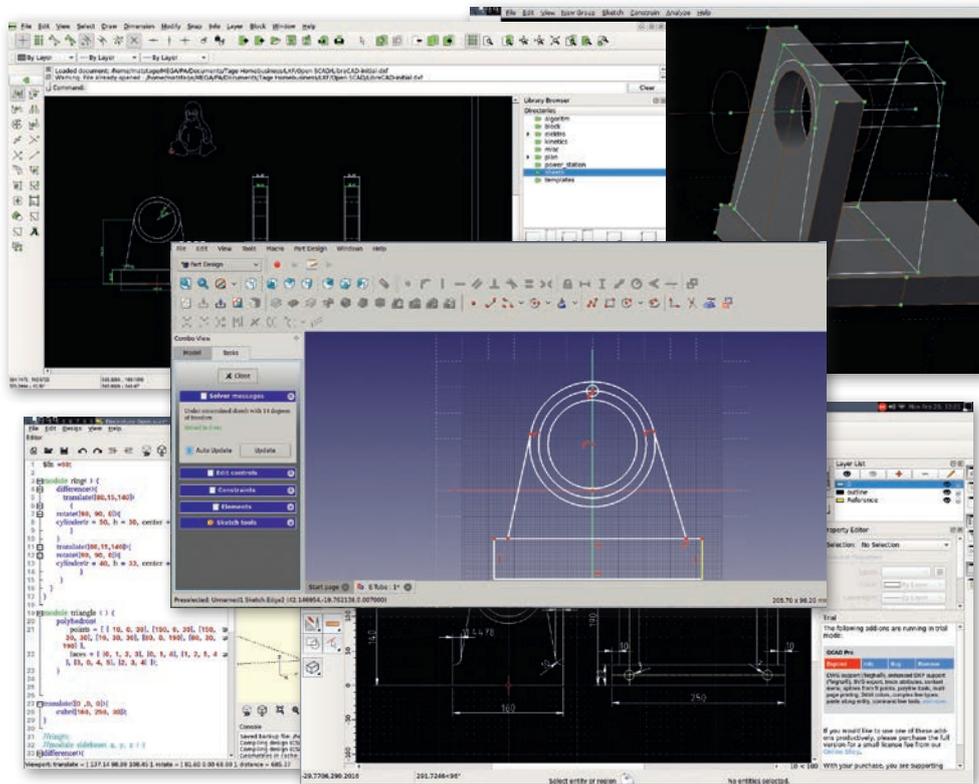
» Рис. 2 (вверху). Нативный проект Tizen.
» Рис. 3 (внизу). Генерация сертификата.

Сравнение

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

CAD для новичков

Какую программу CAD вы возьмете для своего следующего проекта по механике? **Матс-Теге Аксельссон** ведет вас по лабиринту возможностей.



Про наш тест...

Все программы тестировались на двухъядерном ноутбуке с 4 ГБ памяти, на котором стоит Ubuntu 16.10.

В самом начале очень полезно почитать руководства, чтобы разобраться в основах программы и понять, подходит ли она для вашего проекта. Поэтому в нашем тесте мы смотрели, хорошо ли руководства составлены и как быстро мы сможем приступить к работе. Мы также пытались создать одно и то же изделие во всех программах, чтобы выяснить, где мы скорее получим хороший чертеж или конвертируем его в другие форматы.

Процесс установки тоже тестировался на предмет удобства и доступности в дистрибутивах. А еще мы проверяли, что и где можно скачать и загрузить для каждой программы, выложив свои работы в Сеть для общего пользования и повторного употребления.

И, наконец, мы изучили скрипты фигур, полезных для художественного дизайна.

Когда нам надо что-то сделать, мы в *Linux Format* обычно пишем программы, но это не поможет сделать мыльницу для вашего душа. В нашем Сравнении мы исследуем возможности создания разных предметов для физического мира с помощью Linux и Computer Aided Design — CAD. Будучи пользователями Linux, мы, как и всегда, избалованы выбором.

Программы CAD функционируют двумя основными способами, помимо работы в 2D и/или 3D: параметрическим и прямым. Вы можете писать скрипт для создания собственного дизайна или рисовать мышью. Иногда эти два способа сочетаются.

Удобство использования программ CAD зависит от того, что вы хотите создать. Если вы, например, создаете деталь для своего велосипеда, то вам, вероятно, на первый раз стоит взять функцию эскиза и потом задать размеры с помощью размерных линий, чтобы всё получилось правильно.

Если ваши конструкции более замысловатые, то, возможно, окажется проще, или как минимум эффективнее, научиться

работать со скриптами и комбинировать разные фигуры, например, круги, кубы, конусы и т.д. Это бывает довольно сложно, и надо читать пояснения к функциям, иначе ваша конструкция будет бесполезна.

Когда вы наберетесь опыта и будете чувствовать себя со своей программой увереннее, вам надо будет визуализировать свое изделие, чтобы оценить его сильные и слабые стороны.

Удобство использования программ CAD зависит от того, что вы хотите создать.

Наша подборка

- »» FreeCAD
- »» LibreCAD
- »» OpenSCAD
- »» QCAD
- »» SolveSpace

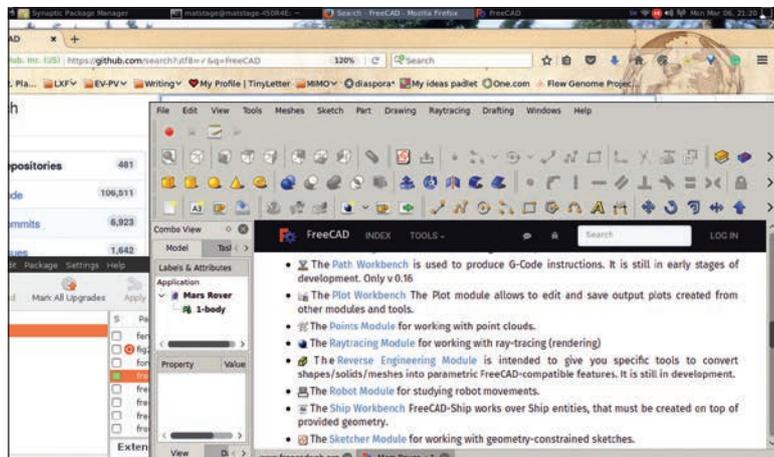
Установка

Большинство программ доступны для Debian или дистрибутивов на RPM.

Хотя установка программ в нашем случае по большей части является вопросом их выбора из репозитория вашего дистрибутива, это не даст вам ничего ультрасовременного и не будет работать с плагинами или библиотеками шаблонов. Если вам нужно нечто ультрасовременное, самое лучшее — освоить Git, поскольку это обычный способ взаимодействия с открытыми репозиториями разработчиков для программистов.

Посмотрим, что происходит. *LibreCAD* v2.0 доступен в репозиториях основных дистрибутивов; также имеются библиотеки с доступными деталями и шаблонами. Они находятся на GitHub, и, поскольку *LibreCAD* работает с файлами DXF, еще больше доступно в Сети, особенно в каталогах запчастей — многие производители запчастей обеспечивают чертежи CAD для своих продуктов. Другие модули не разрабатываются.

FreeCAD в данный момент имеет пакет Debian версии 0.16. Самая новая версия — 0.17, и она доступна только в виде tar или для клонирования с GitHub. На GitHub вы также можете найти много модулей



и готовых частей для расширения вашей оригинальной установки.

SolveSpace имеет версию 2.1 на Debian. Его репозиторий GitHub имеет версию 2.3 в качестве двоичного пакета релиза, и доступен исходник (естественно). Модуль для расширения *SolveSpace* нет, как нет и каталогов специальных запчастей.

У *OpenSCAD* имеется версия 2015.03-1, доступная в Ubuntu 16.04 — с тех пор эта

версия не выкладывалась на их страницу. Для поиска модулей просто используйте Git, чтобы превратить библиотеку MCAD в подмодуль *OpenSCAD*.

QCAD Pro доступен в виде файла скрипта, который устанавливает *QCAD* на вашу систему, а издание сообщества доступно только в виде исходного кода на GitHub. На GitHub имеется модуль механизма, написанный на JavaScript.

» Знание того, что вам нужно — ключ к выбору ваших инструментов.

Вердикт

- FreeCAD ★★★★★
- OpenSCAD ★★★★★
- LibreCAD ★★★★★
- QCAD ★★★★★
- SolveSpace ★★★★★

» В этой категории OpenSCAD набирает несколько очков.

Активность разработки

Уровень активности говорит о сроке жизни программы.

Чтобы выбрать правильный пакет, необходимо знать цель вашего проекта, и основываясь на этом, выбирать, будете ли вы использовать 3D-принтер, фрезерный станок с ЧПУ или другие, более традиционные методы его реализации.

Вам также надо знать, хорошо ли будет поддерживаться пакет и какие новые функции ожидаются в ближайшем будущем. Если вы намереваетесь участвовать в разработке кода, будет небесполезно еще и знать, насколько активно сообщество разработчиков.

Чтобы получить реальное представление о ситуации, мы заглянем на страницу разработки каждого пакета — обычно это GitHub.

LibreCAD имеет на GitHub версию 3, что кажется многообещающим, и его код аккуратно разделен на три части, что говорит о хорошей организации. Имея 47 участников в общем, из которых 7 занимаются версией 3, этот проект может весьма быстро

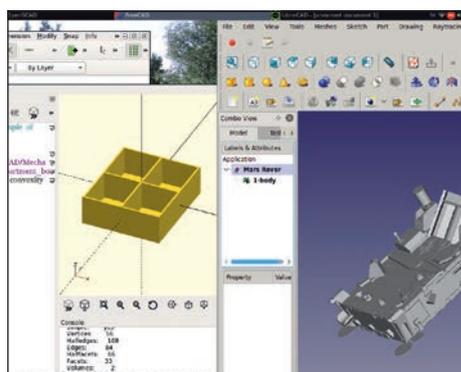
улучшаться. Однако в их планы, похоже, не входит поддержка 3D.

SolveSpace всё еще находится в разработке, но у него очень мало сотрудников. Правда, заявки обрабатываются довольно быстро, так что надежда всё же есть.

FreeCAD пока что в стадии бета, 0.17, но уже вполне пригоден к употреблению, и имеет множество функций, таких, как разбиение объекта на многоугольники и поддержка FEM [модель конечных элементов].

Обработка обусловленности в *sketcher* нуждается в некоторых улучшениях, поиск окончательной реакции периодически бывает утомительным.

Тем не менее, у проекта сильная команда разработчиков и множество подпроектов, и если они будут хорошо координироваться, то перед ними открывается большое будущее. Языком скриптов выбран Python.



» Самая высокая активность у FreeCAD и OpenSCAD, но выбирайте в соответствии со своими нуждами.

Вердикт

- FreeCAD ★★★★★
- LibreCAD ★★★★★
- OpenSCAD ★★★★★
- SolveSpace ★★★★★
- QCAD ★★★★★

» Если вас радует ваш выбор, то помогите в разработке.

Удобство в использовании

Легко ли будет спроектировать свою первую деталь?

Вы быстро понимаете, как много пользы от программы CAD. Примером может служить работа с 3D-принтерами. Если у вас такой есть, вы, возможно, уже использовали какую-то программу для создания чертежей и конвертировали свой файл для использования в своем 3D-принтере.

В данном разделе мы рассмотрим программы, их функции и то, какое влияние всё это окажет на ваш выбор.

Нам надо ответить на следующие вопросы: создаете ли вы чертеж как образец, или для применения в ремонтной мастерской, или вы намерены отправить результаты в другую систему, напри-

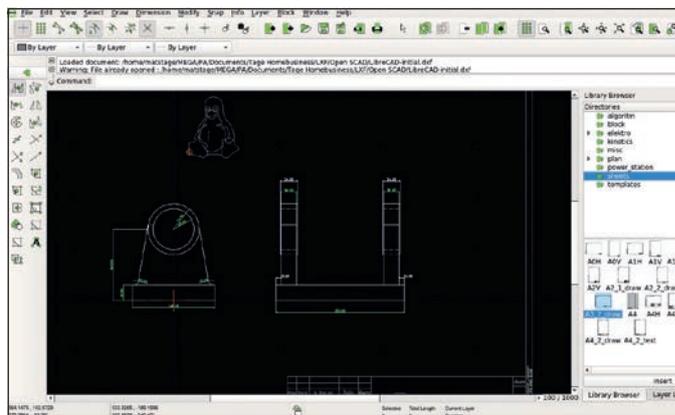
мер, на 3D-принтер либо в программу создания анимации, или, может быть, вам просто надо создать красивую 3D-модель для включения в свою презентацию.

Одни программы предназначены специально для создания чертежей, другие — для моделирования.

LibreCAD ★★★★★

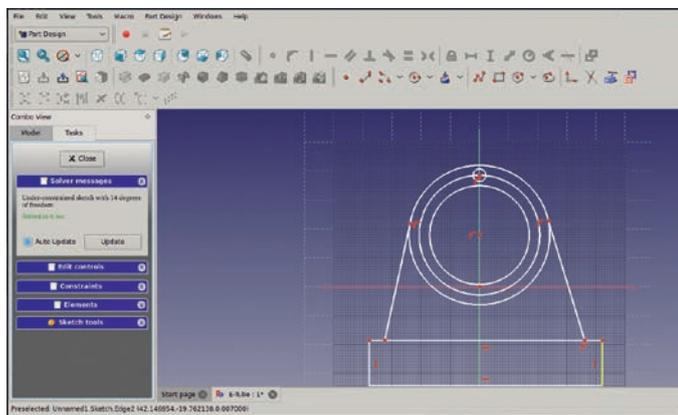
Эта программа создания 2D-чертежей совершенно явно предназначена для нанесения чертежа на бумагу более простым способом, чем карандашом. При ее использовании вам придется с самого начала как следует обдумать то, что вы хотите начертить — включая размеры. Работать с ней немного неудобно, но большинство функций понятны, и изучение действий в командной строке для начала работы и обработки привязок для каждого случая делает программу простой в использовании. Размеры задаются в соответствии с тем, что вы чертите — это немного утомительно, другие программы используют «Режим эскиза [Sketch mode]», который нам показался проще, особенно если вы решите изменить размеры в процессе работы.

Функции конвертирования в 3D нет; самое близкое к ней — изометрический вид. В общем и целом, *LibreCAD* весьма неплох для чертежей, но ему не хватает возможностей моделирования.



FreeCAD ★★★★★

Изначальный вид во *FreeCAD* вообще не связан с чертежами. Вместо этого вам придется начать с выбора того, что вы планируете сделать. В нашем Сравнении мы будем чертить металлический уголок, поэтому мы выбрали опцию «чертеж детали [part design]». Работая с инструментами *FreeCAD*, вы чертите сначала прямоугольники, круги и прямые, и затем устанавливаете требования, которые определяют размер и размещение относительно других частей и линий. В данном случае мы чертим линию от другой линии до касательной к окружности. Этот режим покажет вам, сколько в вашем эскизе степеней свободы; пока это не определено, деталь не получит определенной формы. Это самый сложный шаг в использовании *FreeCAD*. Следующая фаза — создание 3D-модели детали, это делается посредством экструзии. Работа в основном состоит из создания эскиза на поверхности и экструзии для создания целой формы или отверстия.



Сервисы и сообщества

Какие имеются сайты общего пользования и коммерческие сервисы?

Возможность поделиться идеями является частью станового хребта Linux, вместе с принципом свободного и открытого ПО, и это будет так во время ваших приключений в программах CAD. Чтобы иметь возможность делиться своими чертежами, вам надо применять одинаковые программы или одинаковый формат файлов. Сервисы в Сети используют разные форматы в зависимости от своей цели. Сайты для 3D употребляют формат STL, в запчастях от поставщиков используются файлы DXF или DWG, а часто и файлы STL.

LibreCAD использует файлы DXF, и это прекрасно для 2D-чертежей, что и является целью программы. Однако многие поставщики предлагают для ваших проектов 2D- и 3D-чертежи DXF и DWG, так что отсутствие поддержки 3D — это ограничение. *FreeCAD* уже работает со многими форматами, и у вас не должно быть проблем с использованием и созданием рендеров со скачанными частями или эскизами с распределенным доступом. Загляните на сайт 3D-моделирования Thingiverse, где большинство файлов — в формате STL

или STEP, и вы увидите там ярлычок *FreeCAD*. У *SolveSpace* не так много пользователей и нет сообщества для распределенного доступа, а импортируются только файлы DXF, что затрудняет применение чертежей, созданных другими. У *OpenSCAD* есть опция импорта (в скриптах), и можно экспортировать во многие форматы, это упрощает обобществление чертежей. Внутренние форматы *QCAD* — DXF и DWG, поэтому возможность общего использования для 2D-работ очень проста. Файлы также можно добавлять в локальную библиотеку.

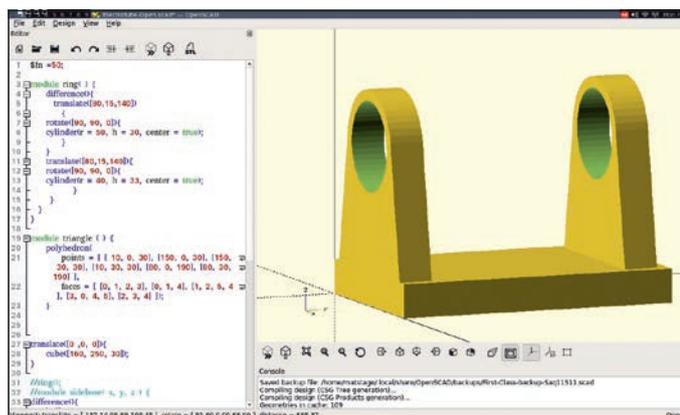
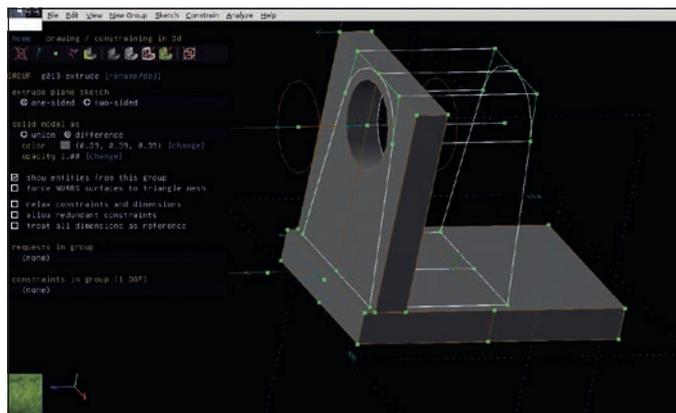
Вердикт

FreeCAD	★★★★★
OpenSCAD	★★★★★
SolveSpace	★★★★★
LibreCAD	★★★★★
QCAD	★★★★★

» LibreCAD и QCAD получают низкую оценку, поскольку они только 2D.

SolveSpace ★★★★★

Плюс *SolveSpace* в том, что часто применяемые функции привязаны к горячим клавишам. Основная работа заключается в создании эскиза в 2D, превращении его в 3D, использовании экструзии и определении размеров и взаимного расположения объектов. Затем выберите на своей детали поверхность, сделайте новый чертеж и выполните экструзию. Интерфейс понятный и очень простой в работе. В долгосрочной перспективе отсутствие функций может оказаться проблемой, более сложные формы будет сложнее создавать, но очень легко начать проектировать. В пользу *SolveSpace* говорят ее функции сборки, простой, но эффективный способ собрать изделие. Вы можете импортировать любой файл, включающий сборку. Это — наилучший выбор, если вы хотите создавать запчасти для замены небольших поврежденных деталей дома. Отсутствие расширенных математических кривых и фигур для большинства пользователей не составит проблемы.

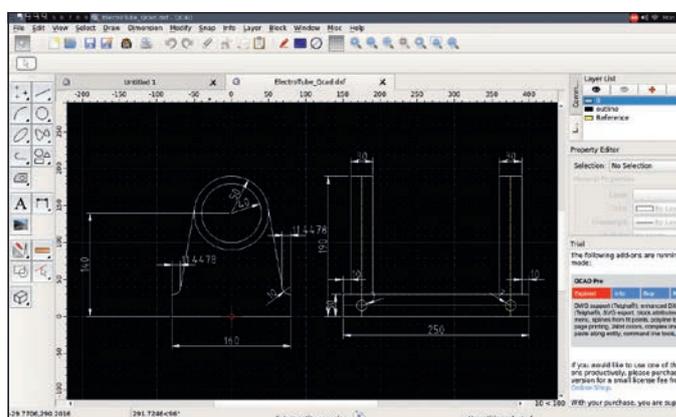


QCAD ★★★★★

В *QCAD* легко приступить к созданию чертежей — функция auto snap [автоматическая привязка к сетке], которая выделяет, какая именно привязка активна, очень понятна. Функции командной строки полезны для создания эскизов первых форм, но для завершения чертежа имеет смысл использовать графический интерфейс. Имеется много функций для копирования, изменений и создания закругленных краев; немало также предопределенных способов для рисования точек, эллипсов, кривых сложного профиля и т.д. Рисование точек может показаться тривиальным, но очень часто они вам нужны в одном или многих определенных местах относительно существующих частей. В *QCAD* нет функций 3D-моделирования, и многие функции экспорта доступны только в профессиональной версии. Функции Pro доступны в течение 15 минут при каждом запуске *QCAD*, потом они прекращают работу. Для любителей, их можно возобновить посредством перезапуска.

OpenSCAD ★★★★★

В *OpenSCAD* единственный способ создания чертежей заключается в создании скриптов, содержащих требуемые формы. Основные формы создавать довольно просто, но если у вас нет опыта программирования, то в создании форм сложнее, чем кубы и шары, у вас будут проблемы. Впрочем, когда вы разрешите эти затруднения, вы сможете довольно быстро создать нужную форму с помощью умных скриптов, а также быстро создавать сложные формы — находить алгоритмы для этого весьма интересно. Здесь отсутствуют встроенные размеры: чтобы их добавить, надо скачать специальные файлы и скопировать их. *OpenSCAD* в основном предназначен для академического использования, но может пригодиться для визуального представления статьи, и если вы привыкли к командной строке, вы сумеете быстро его освоить, ознакомившись с несколькими руководствами и с существующими скриптами и функциями в Сети.



Поддержка и документация

Чтобы приступить к работе, надо изучить документацию и руководства.

Даже если у вас имеется опыт работы с программами CAD, очень важно методично практиковаться, изучая имеющиеся методы и инструменты. Мы рассмотрим, сколько вам придется учиться, чтобы освоить программы в нашем Сравнении.

Качество документации для программы тоже немаловажно, но всё же это не самый практичный способ приступить к работе: руководства помогут вам куда быстрее.

LibreCAD имеет умеренную, но понятную документацию, руководства обычно

понятны и полезны. Есть много руководств на французском, немецком и испанском, но основные вопросы объясняются англоговорящими авторами.

FreeCAD предлагает блестящее решение: вы можете просмотреть их документацию, размещенную в Сети, во вкладке *FreeCAD*, чтобы не надо было открывать для этого браузер. Руководства на YouTube охватывают всё, от основ до создания сеток для сложных функций FEM.

К *SolveSpace* очень легко привыкнуть: ознакомьтесь с одним-двумя имеющимися

руководствами и запишите на бумаге горячие клавиши, и вы будете готовы к работе уже через полчаса.

OpenSCAD имеет огромную библиотеку документации и множество руководств, но будьте готовы к необходимости вспомнить свои навыки в математике. Приступить к работе с этой программой будет несколько труднее и займет несколько больше времени, но у вас будет отличная поддержка в виде документации и руководств.

Справочник *QCAD* краткий и сжатый, без руководств тут не обойтись.

Вердикт

- FreeCAD ★★★★★
 - OpenSCAD ★★★★★
 - LibreCAD ★★★★★
 - QCAD ★★★★★
 - SolveSpace ★★★★★
- » FreeCAD имеет больше руководств, и они лучше, а у SolveSpace их достаточно.

Функции и как вы делаете чертеж

Есть ли поддержка 3D, имитирования и т. д.?

Какие функции требуются вам для вашего проекта? Этот вопрос значительно важнее, чем то, какие функции предлагает определенная программа.

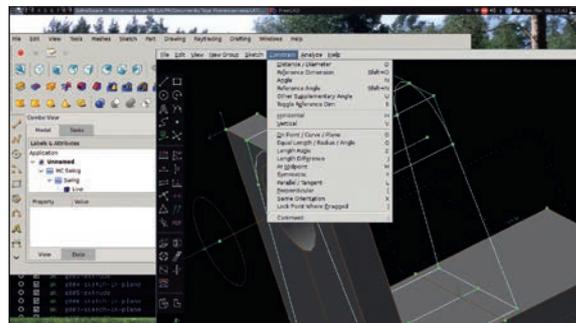
Каждая программа в нашем Сравнении предусматривалась для выполнения определенных задач. *LibreCAD* разработан специально для создания чертежей на бумаге. В нем отсутствуют возможности 3D, тогда как *OpenSCAD* можно применять для создания очень сложных форм и даже 3D-гистограмм; но увы, он сложен в освоении. Многие трудности языка скриптов компенсируются огромным количеством скриптов, размещенных на разных сайтах. Для конструирования или обратного проектирования детали для вашего старого оборудования может быть проще использовать режим эскиза и задать размеры, но если вы создаете новую уникальную работу, использование скриптов дает очень большую мощь.

LibreCAD имеет много библиотек, доступных сразу после установки, а также электросхемы и схемы питания, но гайку только одну. Зато создать собственную

библиотеку очень легко: просто сделайте чертеж с оригинала, сохраните его как DXF и укажите настройкам эту директорию. Вы также можете использовать ту же директорию как библиотеку шаблонов в *QCAD*.

В *QCAD* множество подсказок для работы над разными формами и механическими деталями, что облегчает процесс создания чертежа, но возможности 3D отсутствуют, и вам понадобится отдельная программа моделирования для получения законченной работы, пригодной для вывода на 3D-принтер.

Во *FreeCAD* множество функций для создания эскизов и для их превращения в 3D. Покопавшись в нем, вы найдете модули спиралей и многое другое. Здесь также имеются функции механического анализа, такие как метод конечных элементов (хотя для этого вам потребуется скомпилировать соответствующий модуль), что может быть весьма полезно — в конце концов, перед 3D-печатью вам следует убедиться, что вашей детали не грозит поломка. Набор функций *FreeCAD* — самый полный среди всех.



» Огромный выбор. Не спешите хвататься за самый простой.

Solvespace имеет поменьше функций, но здесь проще приступить к работе, и есть экспорт в DXF, STEP, STL, PDF и другие форматы. Возможность быстро приступить к работе может оказаться весьма кстати, даже если позднее вам придется перейти к более сложной программе.

OpenSCAD может далеко вас завести после изначального боевого крещения. Вам придется привыкнуть к программированию фигур, но если вы уже умеете создавать простые скрипты, вы должны быстро освоиться. Очень удобно наличие расширений для *Vi*, *Emacs* и *Atom*, так что вам не придется редактировать в самом *OpenSCAD*.

Вердикт

FreeCAD	★★★★★
LibreCAD	★★★★★
QCAD	★★★★★
OpenSCAD	★★★★★
Solvespace	★★★★★

» FreeCAD самый полный, но с ним труднее всего начинать.

Взаимодействие и экспорт

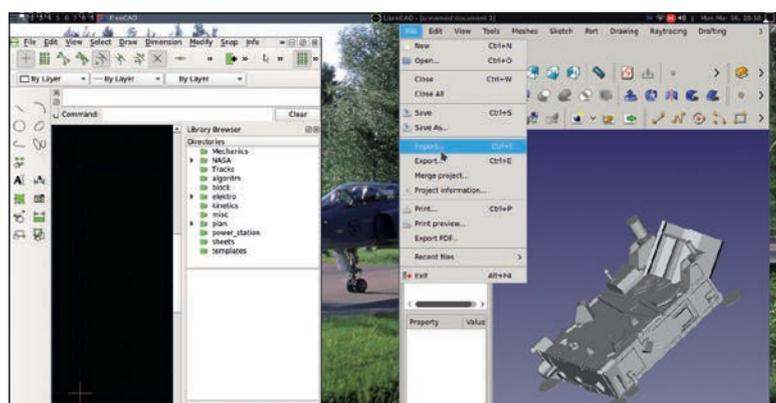
Возможность делиться и преобразовывать — великая радость творчества.

В начале работы в CAD вам нужны руководства и документация, но лучший способ обучения — на примерах от других. И поэтому наилучшим следующим шагом будет посещение общедоступных сайтов для скачивания и изменения работ (не забудьте прочитать лицензионное соглашение работы, которую хотите скачать).

По этой причине очень важно, чтобы вы могли легко импортировать чертежи в свою программу CAD.

LibreCAD экспортирует в очень небольшое количество форматов — DXF, PDF и Image — за что и получает в данном тесте невысокую оценку. Есть поддержка файла ESRI, и *LibreCAD* также может найти точки ASCII в обычном изображении.

FreeCAD располагает впечатляющим набором инструментов для экспортирования и импортирования, что, вместе со структурой плагинов, делает его победителем в этом тесте. Однако будьте осторожны: импорт DXF не так прост, поскольку требует



» Ищите сайты для 3D-печати и чертежей общего пользования.

использования *sketcher*, чтобы превратить его в 3D-структуру в файле.

OpenSCAD использует файлы CSG — или Constructive Solid Geometry. Для скачивания доступно множество чертежей, многие выпущены под лицензией Creative Commons, которая разрешает ремиксы. Импорт PNG или DXF происходит внутри вашего скрипта.

SolveSpace импортирует только DXF, но экспортирует во множество других форматов, что делает его очень удобным в использовании, когда вы создаете собственные творения.

QCAD катастрофически не хватает форматов для импорта и экспорта: поддерживаются только SVG и BMP, PNG и JPG. Но у него отличная библиотека деталей.

Вердикт

FreeCAD	★★★★★
QCAD	★★★★★
SolveSpace	★★★★★
LibreCAD	★★★★★
OpenSCAD	★★★★★

» FreeCAD победил, но вам-то нужен правильный формат, а не их куча.

CAD с открытым кодом

Вердикт

Когда большинство из нас приступает к работе с CAD, мы хотим рисовать формы и детали с помощью мыши или карандаша и даже «строить» их на экране. Это самый интуитивный способ работы, да и стандартный тоже. Процесс усложняется, когда вам нужно приступить к окружностям и кривым Безье, и это только начало. При создании более сложных фигур, особенно если вы хотите создать разницу между двумя фигурами, очень мощным средством может стать использование языка скриптов. *OpenSCAD* использует такой язык, так что не пугайтесь необходимости обучения.

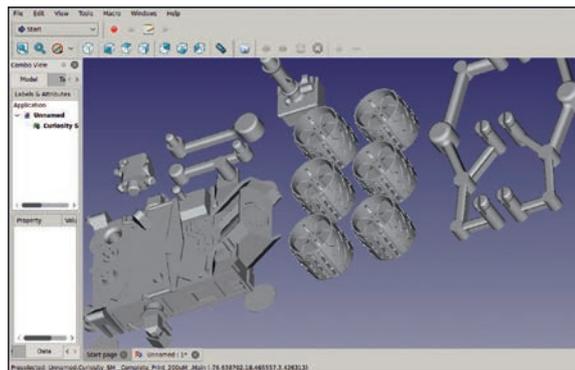
В зависимости от материала, вам может понадобиться в разной степени закруглить углы вашей детали. Обычно эти проблемы решаются после того, как вы начертите квадраты, окружности и углы.

Популярной функцией, лучше всего реализованной в *SolveSpace*, является интерпретатор обусловленности. Вы начинаете с создания нескольких линий

и окружностей, затем задаете их размеры и то, какие линии являются параллельными, вертикальными и т. д. Интерпретатор говорит вам, полностью ли определен ваш чертеж; это означает, что все формы разрешены. Вы также можете увидеть, что ни одна форма не может быть перемещена и не может быть изменен размер, попытавшись перетащить их мышью. *FreeCAD* тоже хорошо с этим справляется.

В *LibreCAD* и *QCAD* вам надо с самого начала правильно чертить фигуры, и когда вы добавляете размеры, они будут показаны именно в этом размере. Вот почему хорошо использовать командную строку для основных фигур и размеров, поскольку изменение размеров впоследствии обычно означает необходимость заново создавать чертеж.

Выбирая одну из этих программ, учитывайте то, что вы будете делать с готовым файлом. Экспорт на сайты определенного доступа обычно



» *Curiosity Rover* предлагается в виде скачиваемого файла для 3D-печати, просмотрите его с помощью *FreeCAD*.

делается в DXF для 2D и STL либо STEP для 3D-принтеров и т. п.

В данном Сравнении мы признаем победителем *FreeCAD*, поскольку с этой программой легко приступить к работе, у нее наилучшие возможности импорта и экспорта и, судя по всему, большое будущее. Однако не стоит игнорировать создание скриптов, оно может оказаться невероятно полезным в вашей работе.

Не стоит игнорировать создание скриптов, оно может оказаться полезным.

I FreeCAD ★★★★★
 Версия: 0.16 Сайт: freecadweb.org Лицензия: LGPL2+
 » Впечатляет своей универсальностью и относительно простым интерфейсом.

IV SolveSpace ★★★☆☆
 Версия: 2.0 Сайт: solvespace.com Лицензия: GPL v. 3
 » Имеет отличный интерфейс, и вы быстро сможете приступить к работе.

II LibreCAD ★★★★★
 Версия: 2.0.9 Сайт: librecad.org Лицензия: GPLv2
 » Очень неплохо работает с 2D, но увы, не располагает функциями 3D.

V QCAD ★★★★★
 Версия: 3.16.5 Сайт: qcad.org Лицензия: GPLv3 (есть проприетарная)
 » Только 2D и очень небольшая поддержка других форматов файлов.

III OpenSCAD ★★★★★
 Версия: 2015.03 Сайт: openscad.org Лицензия: GPL 2.0
 » Сложный процесс освоения, но учитесь, и вам воздастся сторицей.

Обратная связь
 Вы делаете чертежи запчастей для ремонта или создаете детали в 3D? Расскажите нам, какие инструменты вы предпочитаете: lxformat@futurenet.com.

Рассмотрите также...

Не отказывайтесь от web-решений: они постоянно обновляются и добавляют функции. Их масса, и свободные обычно просят вас заплатить только если вам нужны частные чертежи. *Tinkercad* — забавная программа для 3D-моделирования от гиганта CAD, Autodesk. У них также имеются профессиональные сервисы, о которых

мы здесь не упомянули. *Onshape* — отличное облачное приложение, в нем можно создавать полноценные и действительно профессиональные чертежи совершенно бесплатно, с условием, что все ваши чертежи будут общедоступны. Предлагаемые руководства превосходны, и их можно скопировать в виде документа. *Onshape* также задуман

для коллективной работы над проектами онлайн. И есть сайт openjscad.org, где можно редактировать и просматривать результаты языка JSCAD для создания 3D-фигур. Язык — JavaScript, с 95% реализацией языка *OpenSCAD* и применением функций JavaScript. Учетная запись не нужна, и всё можно установить локально, в зависимости от Node.js. **LXF**

Создаем лучшие устройства



Создаем лучшие устройства

Лес Паундер исследует
новый Raspberry Pi Zero W
в качестве платформы
для Интернета Вещей.



Raspberry Pi навсегда изменил то, как проектируются и создаются умные устройства. Когда первый Pi появился на сцене в начале 2012 г., рынок одноплатных компьютеров (SBC, ОПК) предоставлял ассортимент дорогих и сложных устройств разработки пользователей-профессионалов. Raspberry Pi не только снизил цену со \$150 до \$35, но и опустил планку требований к знаниям, необходимым для создания собственных проектов и устройств, благодаря использованию GPL Linux.

В Pi Foundation не стали почитать на лаврах. Спецификации обновлялись и отлаживались. Первое обновление принесло увеличение ОЗУ до 512 МБ. С тех пор мы видели добавление числа контактов GPIO, USB-портов, затем обновление CPU вплоть

до использования четырехъядерного 64-битного процессора, и еще больше ОЗУ.

Pi Zero отступил на шаг назад, предлагая всего 512 МБ ОЗУ и одноядерный CPU с частотой 1 ГГц, но зато это был куда меньший и гораздо более дешевый Raspberry Pi, который можно встроить в проекты и забыть о нем. Pi Zero использовался для проектов камеры, таких, как фотоловушка

было беспроводное соединение.

Новый Pi Zero W вышел 28 февраля, и хотя в нем присутствуют CPU и ОЗУ его предшественника, те-

перь он идет с тем же беспроводным пакетом, что и флагман Pi 3: Wi-Fi 802.11n и Bluetooth 4.1, при умной интегрированной антенне. А значит, теперь у нас есть недорогая платформа для создания приложений Интернета Вещей (IoT) с поддержкой преданного сообщества, которое из почти 12 миллионов Pi, проданных с момента его запуска, насоздало отличные устройства. Вот и посмотрим, как извлечь максимум пользы из этой крошечной платы.

Кое-чего не хватало — того, что мы смогли оценить только с Pi 3 в начале 2016 г.

которые отправлялись на низкую околоземную орбиту и встраивались в носимые технологии. Но кое-чего не хватало — того, что мы смогли оценить только с появлением Pi 3 в начале 2016 г., и это

Творцы будут творить

Почему Raspberry Pi считается главным выбором умельцев?

Своим появлением Raspberry Pi нарушил статус-кво на рынке ОПК. Первый же выпуск в 2012 г. привел к массивному падению цен на платформы разработки. С тех пор мы видели, как многие компании выпускали собственные ОПК в надежде привлечь к себе хотя бы часть интереса, вызванного Raspberry Pi. Огромный диапазон этих продуктов никогда не привлекал такого же внимания, несмотря на наличие в некоторых из них лучших функций, более низкой цены, и т. д. — таких плат, как CHIP, вышедших в середине 2016 г. и имеющих спецификацию, как у Pi Zero.

Однако CHIP предлагал уникальный беспроводной пакет, который включал Wi-Fi 802.11n / Bluetooth 4.0 и встроенное соединение для батарей LiPo, так что ваш проект мог долго работать на батарейке. И всё это за какие-то \$9? Тут же налицо явный подвох?

Конечно, да. То, чего не хватало в CHIP, солидно перевешивало эти функции. В основном ему не хватало поддержки солидного сообщества умельцев, помогающих друг другу изобретать и создавать новые проекты. Кроме того, CHIP не может выводить результат в HDMI без использования специального щитка, блокирующего доступ к контактам GPIO. Так что за эти

начальные \$9 вы не увидите особой пользы для мультимедиа или настольной системы.

На рынке есть и другие платы. Orange Pi, ODROID и Pine предлагают платформу для творчества, но никакие другие ОПК не предлагают той же степени поддержки и такого же сообщества, как у Raspberry Pi.

Как мы видим во флагманских моделях, Raspberry Pi Zero W имеет доступ к выводу в HDMI и воспроизведению видео 1080p HD благодаря GPU VideoCore IV. Поэтому вы можете смотреть свои HD-фильмы на устройстве за £10 или использовать GPIO и запустить воспроизведение видео через огромное количество устройств ввода, возможно, даже для людей с ограниченными возможностями, чтобы они могли использовать устройство без посторонней помощи.

Так что же так привлекает умельцев в Raspberry Pi? Только цена? Отнюдь. Причина в том, что их идеи безграничны. И большие, и малые проекты могут быть созданы на Raspberry Pi. Для всего, от включения обычного светодиода до создания мини-кластеров компьютеров для научных проектов можно использовать один и тот же Pi, и начинающему разработчику потребуются только время и труд для совершенствования своих навыков.

Вкратце

» Это Pi Zero, но с беспроводным соединением. Недорогая и хорошо поддерживаемая отправная точка для проектов Интернета Вещей, робототехники и встраиваемых сенсорных устройств, на той же ОС, что и более дорогие модели.

Raspberry Pi Zero W

Постоянно алчущий нового Pi, на сей раз Лес Паундер получает меньшую порцию, но зато с новыми функциями, а калории на том же уровне Zero.

В конце 2015 г. Pi Zero стал настоящим сюрпризом. Это был новый тип Raspberry Pi, отказавшийся от высоких спецификаций ради создания более дешевой и меньшей по размерам платы. Исходному Zero не хватало одной вещи, но такая же ситуация была и со ставшим

впоследствии флагманом Raspberry Pi 2. Отсутствовало беспроводное соединение.

С выходом в начале 2016 г. Pi 3 мы увидели добавление Wi-Fi и Bluetooth, и сейчас эти функции перепали также и Pi Zero W. Pi Zero W почти идентичен предыдущим моделям Zero. Он идет с тем же портом камеры

(CSI), который появился в V1.3 Zero, и таким же непаянным 40-контактным интерфейсом GPIO (General Purpose Input Output) для подключения Pi к электронным устройствам и добавления плат. Но у Pi Zero W структура платы немного изменилась. Логотипа Raspberry Pi на Zero W уже нет, вместо него — маленький серебряный чип. Это беспроводной пакет BCM4343A, такой же, как в Pi 3. Но вместо смонтированной на поверхности антенны, как в Pi 3, Pi Zero W имеет объемный резонатор, выступающий в роли антенны. Это лицензионная технология от Proant, а не нечто созданное в Raspberry Pi Foundation. BCM4343A позволяет использовать на этой крошечной плате 802.11n Wi-Fi и Bluetooth 4.1 и Bluetooth Low Energy (BLE), делая ее идеально подходящей для Интернета Вещей (IoT).

Рынок IoT видел немало плат, но ни одна не имела такого же уровня поддержки, как Raspberry Pi, который на данный момент продал более 12 миллионов экземпляров

Свойства навскидку



Беспроводное соединение

Пакет BCM4343 такой же, как и в более дорогой Raspberry Pi 3, но на сей раз антенна является частью платы, а не добавляемым к ней компонентом.



40-контактный GPIO

Тот же 40-контактный GPIO, который появился в 2014 г. в B+ и который позволяет использовать Pi Zero W на массивном рынке плат расширения.

Разработка с Pi

С момента выхода Raspberry Pi его официальной операционной системой является Raspbian, и в конце 2016 г. вышла последняя версия под названием Pixel. В ней наблюдается улучшение скорости работы и интерфейса, а также наличие обновленных и новых программ. Для разработки на Python можно использовать Python 2 или 3 через редактор IDLE Python, а те, кто предпочитает более гибкий редактор, могут использовать Geany, общепринятый редактор в сообществе Linux, с рядом плагинов, способных

превратить его в генератор для программирования на разных языках, а не только на Python.

Но Pi не только для программирования. Он может стать низкоэнергетическим сервером для ваших файлов, или web-сервером. Сила Linux в том, что его можно использовать на разных устройствах с совершенно разными спецификациями. Поэтому ваш домашний файловый сервер вполне может быть Raspberry Pi Zero W с подключенным большим жестким диском USB.

Поскольку Raspbian основан на Debian, он относительно прост в установке, и обновляет нужные вам программы с помощью инструмента пакетов APT. Он должен быть вашим первым и главным прибежищем, когда вам нужно будет добавить программы даже для библиотек Python. Но если требуемая библиотека недоступна через APT, вы можете использовать PIP, менеджер пакетов Python, для установки новых библиотек. Он тоже запускается в терминале Linux.

С новым Pi Zero W мы увидим больше встраиваемых проектов с низким энергопотреблением, использующих беспроводное соединение для отправки и получения данных.

Проекты, которым не требуется монитор, могут работать без мыши, монитора и клавиатуры через локальное или удаленное соединение. Проекты «гражданской науки», которые собирают данные с помощью бюджетных датчиков и затем загружают данные в большие репозитории для последующего анализа, могут применять Pi Zero W с большой пользой для себя. И в результате можно будет собрать большее количество информации об окружающем нас мире, причем всё это — благодаря компьютеру за £ 10 размером не больше коробочки с жевательной резинкой.



➤ Структура Pi Zero не очень изменилась с момента его появления в 2015 г., но заметно, что в этот крошечный пакет вложено намного больше.

по всему миру. Zero W идеально подходит на роль стартовой точки для множества людей, чтобы они сделали свои первые шаги в IoT, и это устойчивая экосистема для будущих продуктов.

Pi Zero W также годится и в качестве настольного компьютера. На нем можно запускать LibreOffice для создания офисных документов, он умеет воспроизводить видео 1080p и работать в Интернете. Кроме того, Zero W можно применять в школе для обучения языкам программирования, таким, как Python, Ruby и JavaScript.

Но подлинное достоинство Pi Zero W в том, что это — бюджетный встраиваемый компьютер, который обеспечит проектам беспроводное соединение. Используя Pi Zero W как устройство без монитора, мыши и клавиатуры с доступом через SSH или серийный консольный интерфейс, мы сможем получить доступ ко всей мощи

Pi Zero W, который станет универсальным устройством для робототехники, камер для съемки природы и в качестве платформы обработки сигналов датчиков. Кроме того, Pi Zero W исключительно экономичен в плане энергии: в нашем недавнем интервью (которое вы увидите через несколько страниц) Эбен Аптон сказал, что блока питания USB мощностью 600 мА более чем достаточно для Pi Zero W, даже с активным Wi-Fi. Это намного меньше блока питания на 2,5 А, требуемого для Raspberry Pi 3.

Так на что же нам рассчитывать за £ 10? Это прочная как скала платформа, которая вновь ломает представления о том, на что способен дешевый одноплатный компьютер. Raspberry Pi когда-то привел к снижению цен на платформы разработки, и Pi Zero W делает это снова. Пускай у него и нет четырехъядерного CPU, 1 ГБ ОЗУ и множества USB-портов. Зато вы получаете



➤ По размеру — почти что с коробочку жевательной резинки, но его новые функции беспроводного подключения предлагают впечатляющую платформу.

бюджетную платформу для создания собственных встраиваемых проектов для IoT и носимых технологий. А всего лишь за £ 10 вы можете себе позволить рискнуть и попробовать создать нечто выдающееся.

Сравнительные тесты

Модель	Время загрузки [с]	Sysbench: 1 ядро, 10 000 прогонов	Sysbench: 4 ядра, 10 000 прогонов	Octane	ОЗУ	Процессор
Raspberry Pi Zero W	58	354	N/A	456	512 МБ	1 × ARM 11 1 ГГц
Raspberry Pi Zero v1.3	57	353	N/A	458	512 МБ	1 × ARM 11 1 ГГц
Raspberry Pi 3	31	182	45	2686	1 ГБ	4 × ARM Cortex A53 1,2 ГГц
Raspberry Pi 2	31	243	61	2131	1 ГБ	4 × ARM Cortex A7 900 МГц
Raspberry Pi 1	71	512	N/A	N/A	512 МБ	1 × ARM11 700 МГц

LINUX FORMAT Вердикт

Raspberry Pi Zero W

Разработчик: Raspberry Pi Foundation
Сайт: www.raspberrypi.org
Цена: £ 9,60

Функциональность	9/10
Производительность	5/10
Удобство в работе	9/10
Оправданность цены	10/10

➤ Бюджетный и с хорошей поддержкой. Выберите проект и язык — и бегом за этой платой.

Рейтинг **10/10**

Больше радости от Raspberry Pi Zero W

Добавив HAT на свой Raspberry Pi Zero, вы можете продолжить свои приключения. Лес Паундер изучает пять лучших.

Когда в 2012 г. появился Raspberry Pi, долгие месяцы не было никаких плат расширения. Первой из них стала Gertboard, разработанная местным инженером Гертом ван Лоо (Gert van Loo). Это была массивная плата, предназначенная для расширения GPIO и обеспечивающая аналого-цифровой ввод, буферизованные контакты, контроллер мотора и совместимый с AT Mega микроконтроллер.

Как первенец, Gertboard произвела сильное впечатление, и мы вскоре стали прозывать больше возможностей для Raspberry Pi. Компании вроде Pimoroni, Syntech, 4Ttronix и других стали выпускать собственные платы, все предназначенные для стандартного на тот момент 26-контактного GPIO, и рынок просто взорвался устройствами, способными помочь кому угодно в чем угодно. Первые платы предлагали светофор, простые контроллеры моторов и защищенные контакты GPIO, и все они были предназначены для быстрой и легкой реализации самых разных проектов. Но каждая плата также создавала ряд технологических сложностей, самой заметной из которых был размер устройств, способный

варьироваться от макетных плат до многоуровневых коммутационных плат, таких, как Gertboard. Нужен был стандарт.

Прокрутим до момента появления V+ в 2014 г., и здесь мы видим новый стандарт для плат расширения: HAT, Hardware Attached on Top, задавший размеры, которых обязана придерживаться плата, чтобы размещаться на Raspberry Pi и удерживаться на месте 40-контактным GPIO и отверстиями для болтов на Pi. Также предполагалось, что для общения с Raspberry Pi плата будет использовать EEPROM через несколько новых контактов на 40-контактном GPIO. Стандарт HAT не был официальным, но если вы хотели называть свою плату HAT, вы должны были выполнять его требования.

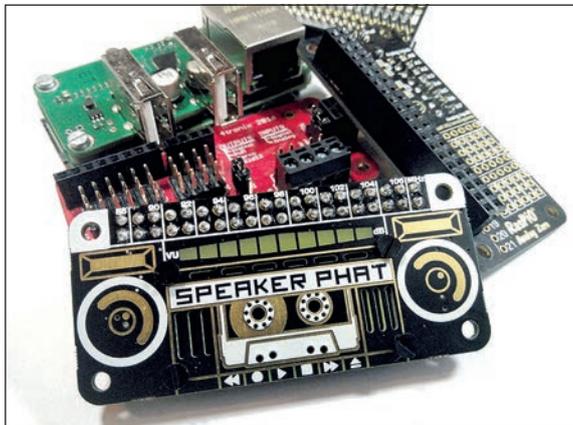
В реальности плату расширения может сделать любой... да почти все так и делали. Есть также стандарт pHAT, разработанный Pimoroni для плат, предназначенных для работы с Pi Zero. Этот стандарт привел к разработке множества новых, меньших по размеру плат, и разрабатывались они не только в Pimoroni. Стандарт открыт для использования, и сейчас мы видим, как многие члены сообщества используют его для своих самых последних плат.

Сейчас, в 2017 г., имеется множество плат расширения, от простеньких разработанных сообществом приспособлений до предельно собранных продуктов массового производства для домашней автоматизации, высококачественного аудио или вообще исследования космоса.

Прелесть плат расширения в том, что они вдохновляют пользователя. Конечно, заставить замигать светодиод весьма забавно, но куда интереснее создать робота и узнать, как управлять моторами на основе ввода информации с датчиков.

В нашей статье мы рассмотрим пять плат, которые обеспечивают дополнительные функции для вашего Raspberry Pi Zero, или других 40-контактных GPIO Raspberry Pi. Мы узнаем, как добавлять звук к проекту любого размера и как встраивать аналоговый ввод для датчиков и компонентов. Затем мы рассмотрим платы, помогающие создать робота микроразмера, цифровое световое шоу или путешествовать в прошлое... или нечто вроде того.

➤ Есть множество плат расширения Raspberry Pi, и все можно применять для создания роботов, сбора данных об окружающей среде или хоть игры на пианино.



Speaker pHAT www.pimoroni.com

В обычном Raspberry Pi мы слышим звук через кабель, подключенный к HDMI или 3,5-мм аналоговому порту. А если вы хотите создать меньший по размеру аудиопроект с помощью Pi Zero? Что ж, теперь это возможно с помощью Speaker pHAT.

Speaker pHAT — простой в сборке конструктор, который потребует всего несколько минут с отверткой и потом еще несколько минут с паяльником. Довольно милая плата, которая выглядит, как Бум-бокс 1980-х, размещается поверх всего 40-контактного GPIO. Программы устанавливаются через простой скрипт установки, запускаемый из терминала.

По завершении установки надо поскорее перезагрузиться, чтобы услышать музыку своего проекта. Не стоит рассчитывать на качество THX, от которого задрожат стекла; звук чистый, но ему не хватает басов, в основном из-за размера платы. Отличной бонусной функцией является наличие на плате простого волюметра для отображения уровня громкости, так что теперь вы можете «видеть» свою музыку, когда она играет. Это интересный проект для того, чтобы привнести звук в ваш проект Zero, будь то подушка-пердушка с дистанционным

управлением или бытовое устройство на основе Alexa. Хотя басов и не хватает, но производится четкий чистый звук.



Analog Zero rasp.io/analogzero

Одно из самых редко замечаемых различий между Raspberry Pi и Arduino в том, что у Pi нет аналогового интерфейса. Таким компонентам, как потенциометры, регуляторы и т.д., нужен аналоговый интерфейс, чтобы корректно сообщать свое состояние, в противном случае они могут только сообщить, включены они или выключены, как в цифровом интерфейсе. Analog Zero обеспечивает мост между аналоговым и цифровым миром, через чип MCP3008. Этот чип считывает состояние аналогового ввода, которых может быть до восьми, и затем конвертирует его для отправки на стандартные цифровые контакты GPIO.

Плата Analog Zero поставляется в виде комплекта сделай-сам, однако на сборку нужно очень мало времени, хотя опять придется вытащить паяльник. Плата также предоставляет небольшую область прототипирования на случай, если вы захотите встроить ее в проект.

Программная сторона отличается необычайной простотой. По сути, вам не требуется ничего устанавливать. Программа для этой платы встроена в GPIO Zero, собственную библиотеку Python от Raspberry Pi Foundation. Вам надо только настроить свой проект на использование этой платы через интерфейс GPIO Zero и затем начать работать

с новыми аналоговыми компонентами.

Плата Analog Zero пригодна для использования со всеми моделями Raspberry Pi, поставляемыми с 40-контактным GPIO. Она удовлетворяет требованиям к плате pHAT, поскольку повторяет контур платы Pi Zero.



Picon Zero www.4tronix.co.uk

Есть ряд проектов, с которых может начаться ваш путь к превращению в умельца. Включение ЖК-дисков; использование WS2812B LED (Неорixel); работа с датчиками; управление мотором для робота.



Так случилось, что Picon Zero как раз умеет делать всё это на одной плате.

Picon Zero поставляется с двумя полноценными контроллерами моторов с H-мостовой схемой управления (DRV8833), используемыми с микропередачей или со стандартными моторами постоянного тока для разработки подвижных проектов. Питание моторов может осуществляться через Pi или от внешнего источника питания, что очень удобно, если у вас мощные моторы.

Выходы могут использоваться как обычные контакты GPIO или для управления сервомашинкой. Ввод можно контролировать как аналоговый, цифровой или используемый с температурным датчиком (DS18B20) без использования нагрузочного

резистора 4,7 кОм. Точно так же можно присоединить к плате ультразвуковой датчик HC-SR04, обеспечив робототехнический проект датчиком расстояния, не требующим делителя напряжения.

Эта плата является отличной платформой для создания проектов, особенно в области робототехники. Она работает со всеми платами Raspberry Pi от V+ и выше и предназначена для размещения на Raspberry Pi Zero.

Это бюджетная платформа с уменьшенным трением для создания физических компьютерных проектов на Python. Плата поддерживает столько разных частей, таких, как датчики, моторы, приводы и светодиоды WS2812B, что позволит даже новичкам сделать первые шаги.

PiBorg ZeroBorg www.piborg.org

PiBorg создают роботизированные решения робототехники «для больших». Но для ZeroBorg они создали маленькую плату pHAT, с независимым контролем до четырех моторов и общим максимальным током 2 А. Это солидная мощность для платы подобного размера.

И кстати о мощности: эта плата питается и может также запитать Raspberry Pi Zero от батарейки 9 В, что позволяет создавать очень маленьких, но мощных роботов. Если вы дожидались поры создавать армию неуправляемых киборгов, то она настала.

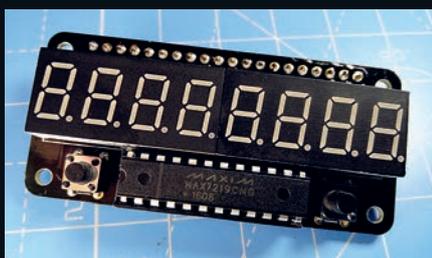
Плата поставляется с двумя аналоговыми вводами, используемыми для датчиков, и общается с помощью I2C, который задействует всего два контакта на GPIO для полноценной коммуникации плат.

Имеется две модели ZeroBorg, KS1 и KS2. У KS2 больше всего функций, и именно ее мы рассматривали. Она поставляется с инфракрасным датчиком, пригодным для управления роботом с помощью пульта для телевизора, а также предлагает интерфейс шлейфового подключения I2C, позволяя присоединить другие платы.



Вам нужно будет кое-что спаять — если быть точными, шесть контактов GPIO; и это создаст полупостоянное соединение между Pi и ZeroBorg. Если вы серьезно увлечены робототехникой, вам понравится эта плата!

ZeroSeg www.thepihut.com



В конце XX века 7-сегментный индикатор был крутой штуковиной, а ныне, в XXI веке, мы можем собрать его сами.

Скромные 7-сегментные индикаторы были повсюду в 1980-х и 1990-х. Они сообщали время на нашем видеомониторе, постоянно мигающие 12:00. ZeroSeg — это плата, совместимая с Raspberry Pi Zero, которая предлагает два четырехзначных 7-сегментных индикатора, управляемых чипом MAX7219CNG с помощью шины SPI. Ее комплект сборки требует наличия твердой руки и логического мышления, он также снабжен двумя тактильными выключателями для ввода, которые легко запрограммировать. И это приводит нас к программам. Плата использует библиотеку Python 2, к сожалению, не Python 3, но она действительно проста

в использовании. Через пару минут на нашей плате появилось прокручиваемое сообщение, затем мы написали простой скрипт для прокручивания температуры в нашем родном городе. Эта плата была разработана как простой способ вывода информации для пользователя и для еще более простого ввода.

ZeroSeg — симпатичная плата, созданная членом сообщества для сообщества. Собрать ее немного сложно, но если вы будете следовать инструкциям, вы отлично проведете время, возрождая небольшой кусочек истории, скромный 7-сегментный индикатор.

Эбен Аптон

Мы расспросили босса Pi о новом Zero и о том, что будет дальше.

Linux Format: Здравствуйте, Эбен, спасибо, что уделите время на беседу с LXF. Итак, только что вышел новый Raspberry Pi Zero W, но зачем нам еще один Raspberry Pi?

Эбен Аптон (Eben Upton): Мы видели, что очень многие подключали к своим Pi Zero приставки Wi-Fi и Bluetooth. Когда мы впервые выпустили Pi Zero в 2015 г., у нас не было своих беспроводных продуктов на рынке, поскольку мы всё еще поставляли Raspberry Pi 2 и готовили к выпуску Raspberry Pi 3. Урок, который мы получили, создав беспроводной пакет для Raspberry Pi 3, был усвоен и добавлен к новому Pi Zero W. На то было две причины: во-первых, что очевидно, вам больше не нужна приставка Wi-Fi, и во-вторых, благодаря использованию Bluetooth люди смогут подключать клавиатуры и мыши Bluetooth к своему Zero, не нуждаясь в USB-хабе. Поэтому сейчас мы подошли к тому, что Zero W стал более законченным настольным компьютером.

LXF: Уже проводятся сравнения между исходным Pi Zero и компьютером CHIP от Next Thing Co, компьютером за \$9 с такими же спецификациями, как у Zero, но предлагающим Wi-Fi и Bluetooth в качестве стандарта. Чем Zero W лучше?

ЭА: У Raspberry Pi намного лучше воспроизведение мультимедиа, способное показывать видео 1080p через HDMI по умолчанию через порт мини-HDMI, тогда как CHIP требует коммутационной платы HDMI, которая к тому же загромождала контакты GPIO. CHIP — конкурентный продукт, и нам нравится форм-фактор. Конечно, «каждому свое», и для некоторых проектов CHIP может быть лучше, но нам кажется, что Pi Zero W — лучшая общецелевая плата.

Нам (Raspberry Pi) нравится, что есть другие платы. До того, как мы начали выпускать Raspberry Pi, все одноплатные компьютеры стоили \$125–150. Но с момента выхода Raspberry Pi цены резко упали, и мы гордимся тем, что дошли до такого уровня, когда можно выбрать плату в пределах \$10.

LXF: Когда вы начали исследовать возможности Pi Zero W, были ли у вас другие соображения по поводу улучшений или функций?

ЭА: Я думаю, для Pi Zero W мы сделали, что могли, у нас нет альтернативного более быстрого процессора. Естественно, у нас есть чипы Raspberry Pi 2 и 3 (BCM2836 и BCM2837 соответственно), но ни один из них не подходит под форм-фактор Raspberry Pi Zero W. Когда мы впервые выпустили Pi Zero, мы умудрились втиснуть 1 ГГц от BCM2835, что было шагом вперед по сравнению с исходным оборудованием Raspberry Pi.

LXF: Это уже третья модель Pi Zero; исходная модель вышла в ноябре 2015 г., затем в начале 2016 г. добавился интерфейс камеры. Есть ли планы постепенно снимать с производства старые модели?

ЭА: Исходной версии без подключения камеры на самом деле уже нет, мы не изготавливали ее в большом количестве, и она считается «межеумочным» Raspberry Pi Zero. «Настоящий» Raspberry Pi Zero — это версия с интерфейсом камеры. Эта версия останется в производстве вместе с новым Pi Zero W. С Raspberry Pi Zero всё сурово, они по-прежнему отпускаются по штуке на человека, но доступны для приобретения. Было бы неправильно выпускать новый Raspberry Pi Zero W, не обеспечив хорошего запаса исходных. Интересно посмотреть, сколько людей по-прежнему будут выбирать версию за \$5, предпочитая ее Pi Zero W, ведь за лишние \$5 вы получаете намного больше функций.

LXF: Будет ли в ближайшее время увеличено количество моделей Pi Zero?

ЭА: Это не первоочередная задача. Сейчас, когда всё выглядит довольно стабильным, наша задача заключается в том, чтобы производить достаточно для создания запаса. У нас всегда была эта проблема: мы выпускаем новый продукт, он нравится людям, они принимают его покупать — и продукт заканчивается. Поэтому наш приоритет — обеспечить производство достаточного количества изделий, чтобы они были в наличии, причем в большем количестве мест. Так что вы увидите больше Pi Zero W в большем количестве магазинов в Германии, Франции, Нидерландах, Японии и в США. Мы прилагаем усилия к расширению ассортимента Pi Zero, но будем следить и за увеличением количества единиц, которые можно приобрести, возможно, отменяя максимальный предел.

LXF: Будет ли в Великобритании больше фирм с запасами готовой продукции, таких, как Premier Farnell или RS Components?



ЭА: На первых порах мы на это не рассчитываем. Просто обращаем на это внимание время от времени. Для них это не главный бизнес, и мы пока не достигли того уровня, когда для них это станет разумно с коммерческой точки зрения. Но мы постоянно будем отслеживать ситуацию и посмотрим, куда мы придем.

LXF: Итак, розничная цена Pi Zero W — около \$10? Весьма конкурентоспособная цена.

ЭА: Да, цена примерно £9,60 с НДС. Если бы в детстве мне сказали, что у меня может быть такая машина меньше чем за £10, я бы решил, что это вранье. В детстве у меня был Amiga 600 с 1 МБ ОЗУ и 7 МГц CPU, и он стоил пару сотен фунтов. Сегодня я могу купить компьютер с 1 ГГц и половиной гигабайта памяти меньше чем за £10! У Raspberry Pi богатое наследие, как у Amiga... для меня это — современный Amiga, поскольку Raspberry Pi, как и Amiga, имеет индивидуальное оборудование в серии чипов BCM283X, индивидуальные 3D-ускорители и т.д. Только 2–3% чипов BCM283X предназначены для процессора ARM, остальные состоят из разных ускорителей (2D, 3D и т.д.).

LXF: За последние годы возрос интерес к такому явлению, как Интернет Вещей, и вот у нас есть дешевая платформа, готовая к употреблению в Интернете Вещей. Принимали ли вы в расчет IoT, начиная проектировать Pi Zero W?

ЭА: Безусловно. Я думаю, это будет отличная платформа, по крайней мере, для прототипирования. Это отличный чипсет, оптимизированный для IoT. BCM4343A — отличный беспроводной чипсет, и мы ждем увидеть всё, что создадут с его помощью. Мы уже видели, какие потрясающие вещи создаются на исходном Pi Zero с приставкой Wi-Fi, и будет здорово увидеть, что смогут сделать с помощью платформы, которая включает это в качестве стандарта.

LXF: С Wi-Fi и Bluetooth, не увеличится ли энергопотребление нового Pi Zero W?

ЭА: Небольшое увеличение энергопотребления есть, около сотни милливатт. Но это почти не влияет, если вы пытаетесь найти источник питания, не способный обеспечить питание для Pi Zero W. Источники питания 500–600 мА для Pi Zero W хороши. Включение Wi-Fi в Pi Zero W не влияет на требования к питанию, в отличие от Raspberry Pi 3, которому нужен очень весомый источник питания.

LXF: Одна из незаслуженно обойденных любовью моделей Pi — Model A. Увидим ли мы некие улучшения этой модели?

ЭА: А+ — отличная модель, и один из огромных плюсов в том, что она всегда доступна и не ограничена по количеству. Чтобы понять, для чего подходит А+, нужно понимать рынок Raspberry Pi. Вам знакомы образовательный и любительский рынки и промышленный рынок. Говоря о Raspberry Pi, мы, как правило, подразумеваем два первых рынка. Однако промышленный рынок для Raspberry Pi тоже важен, и на нем А+ является исключительно успешным продуктом. В основном причина в том, что промышленные клиенты могут закупать большие партии в таких местах, как RS Components и Premier Farnell. Периодически мы слышим, что запас А+ закончился и люди

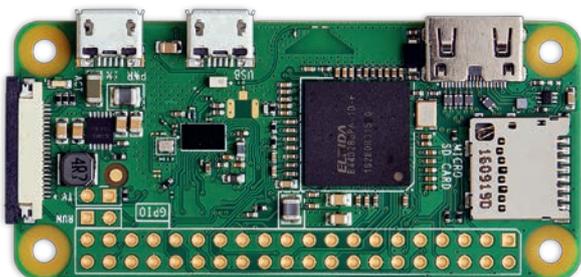


начинают волноваться, не означает ли это конец платы, но причина в том, что клиенты приобретают партии по 10 000–20 000 единиц. Промышленные заказчики обычно разрабатывают свои приложения и продукты на Raspberry Pi 3, а потом ищут возможность втиснуть свои продукты на А+.

У меня ужасный опыт предсказаний про покупателей конкретного продукта.

LXF: Стоит ли рассчитывать на переход каких-то функций Pi 3 в серию А? Может быть, 3А?

ЭА: Я полагаю, что не хватает линии плат 2А/3А. У нас нет конкретного плана, и сделать 3А сравнительно просто, но совместимость (для беспроводного радио) довольно дорога, поскольку мы должны искать совместимость в каждой стране, где есть продажи. И кому это надо? Я думаю, рынок любителей будет заинтересован так же, как они заинтересованы в серии Zero. Включение Wi-Fi и Bluetooth расширит использование единственного USB-порта. Я представляю, что 3А будет промышленным продуктом, но обычно говорю, что это неправильно, у меня ужасный опыт предсказаний про покупателей конкретного продукта. По-моему, 3А мог бы вызвать интерес у любителей, но пока все BCM2837 используются для производства Raspberry Pi 3, поскольку спрос на этот продукт просто зашкаливает.



Создаем сенсорную сеть IoT нового поколения

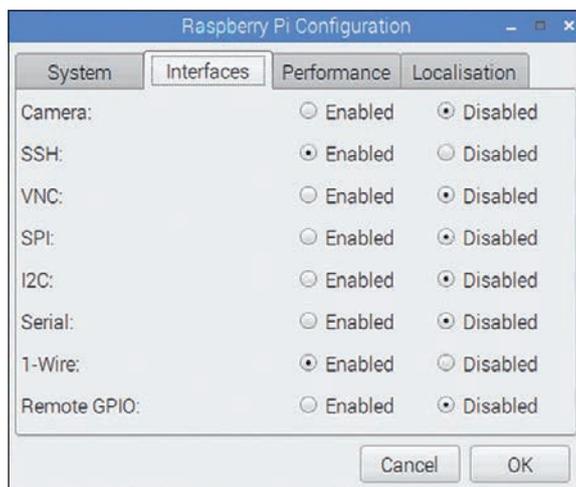
Лес Паундер представляет облегченный протокол обмена сообщениями, пригодный для передачи данных в любую точку мира.

Зрелая экосистема IoT требует стандартов. Знакомьтесь: MQTT, межкомпьютерная прикладная платформа. С помощью модели Publisher/Subscriber [Издатель/Подписчик] данные можно передавать от Издателя, используя темы, и они будут отфильтрованы по темам у Подписчика, который потом может реагировать предсказуемым образом.

Мы создадим сеть MQTT. Издатель, Pi Zero W, будет отправлять данные о температуре через сеть MQTT, где Подписчик, другой Pi Zero W, будет отвечать сменой цвета светодиода (LED) на плате Blink.

Наше оборудование распределено между двумя платами Pi Zero W. Плата Издатель имеет температурный датчик DS18B20, подключенный в соответствии со схемой проекта. После подключения загрузите на Pi-Издатель рабочий стол Pixel, из главного меню перейдите в Preferences и затем нажмите на Raspberry Pi Configuration, чтобы загрузить приложение. Открыв его, нажмите на вкладку Interfaces и включите интерфейс 1-Wire, после чего перезагрузите Pi. Итак, мы успешно включили интерфейс для нашего датчика температуры DS18B20.

У платы Подписчика просто подключена к GPIO плата Pimoroni Blink LED.



➤ Благодаря приложению Raspberry Pi Configuration настройка серийного подключения 1-Wire для нашего DS18B20 проходит со свистом.

➤ Посредник [Broker]

Мы начинаем с настройки Raspberry Pi, который будет выступать в роли Посредника между Издателем и Подписчиком. Откройте Терминал и установите сервис Mosquitto (MQTT):

```
$ sudo apt update && sudo apt install mosquitto mosquitto-clients
Также запишите IP-адрес Посредника, узнав его с помощью
$ hostname -I
```

➤ Издатель [Publisher]

Издатель — это Pi Zero W, который отправляет данные, собранные от датчика температуры DS18B20, подключенного к GPIO, Подписчику Pi Zero W.

Откройте терминал и обновите список программных пакетов, затем скачайте библиотеку Python, которая позволит датчику общаться на Python. И, наконец, установите библиотеку MQTT для Python 3:

```
$ sudo apt update
$ sudo apt install python3-w1thermsensor
$ sudo pip3 install paho-mqtt
```

После установки мы можем перейти к написанию кода Python 3 для этого проекта. Для этого воспользуемся Python 3 Editor, он находится в меню Programming. Когда редактор откроется, нажмите на File > New [Файл > Новый], чтобы создать пустой документ. Далее нажмите на File > Save [Файл > Сохранить] и назовите файл **MQTT-Publish.py**.

Начните с импорта библиотек, первой из которых будет библиотека *time* для контроля за интервалом между считываниями с датчика. Далее идет библиотека MQTT, а затем библиотека датчика DS18B20.

```
import time
import paho.mqtt.client as mqtt
from w1thermsensor import W1ThermSensor
```

Затем мы создаем две переменных, которые будут использоваться в качестве объектов для нашего датчика и соединения с MQTT.

```
sensor = W1ThermSensor()
client = mqtt.Client()
```

Следующая строка кода подключит Издателя к Pi-Посреднику. Укажите IP-адрес Посредника, который вы записали ранее. Нам надо также указать порт, куда будут отсылаться данные, и, наконец, длительность поддержки соединения в состоянии активности, в секундах.

```
client.connect("BROKER IP ADDRESS", 1883, 60)
```

Используйте цикл `while True:` для получения текущей температуры от датчика, затем сохраните ее в виде переменной. Мы назвали ее `hall`, поскольку датчик был в прихожей. Далее мы публикуем данные с помощью MQTT, публикуя данные в теме `temp` и затем передавая переменную `temp` с данными температуры. Затем печатаем данные в оболочке, для целей отладки. Потом засыпаем на 10 секунд перед повторением цикла. В реальном применении этот сон может быть намного дольше.

```
while True:
    temp = sensor.get_temperature()
```

Вам нужны

- 2xRaspberry Pi Zero W
- 1xRaspberry Pi (любая модель)
- Подключение к роутеру (проводное или беспроводное) для каждого Pi
- Pimoroni Blink
- Температурный датчик DS18B20 в водонепроницаемом рукаве

- Резистор 4,7 кОм (желтый, фиолетовый, красный, золотой)
- Блок терминала (из магазина DIY)
- Свежий образ Raspbian/Pixel
- Весь код для данного проекта и коммутационная схема находятся на <https://github.com/lesp/LXF223-PiZeroW-MQTT/archive/master.zip> (<https://bit.ly/LXF223mqtt>)

Что такое MQTT?

MQTT — сокращение от Message Queue Telemetry Transport; он работает как облегченный протокол передачи сообщений для использования поверх протокола TCP/IP. MQTT — идеальный выбор для устройств с ограниченной пропускной способностью сети, поскольку его можно использовать даже при соединении с Интернетом через телефонную линию. Первую версию MQTT создали в 1999 г. Энди Стэнфорд-Кларк [Andy Stanford-Clark] и Арлен Ниппер [Arlen Nipper], и с тех пор она используется в проектах домашней автоматизации, сетях датчиков и для управления гирляндами на новогодних елках по всему миру.

Как мы упоминали в начале руководства, MQTT — это межкомпьютерный протокол, который позволяет устройствам публиковать и подписываться на интересные «темы». В нашем проекте мы создали небольшую сеть, где один Издатель отправляет данные Посреднику, а тот распространяет сообщения клиентам, которые называются Подписчиками и реагируют на определенные темы данных.

MQTT не ограничивается Python; на самом деле, установив пакеты MQTT для Linux, мы можем отправлять и получать данные в терминале. Чтобы установить MQTT, откройте терминал и введите

```
$ sudo apt install mosquitto mosquitto-clients
```

Итак, если вы хотите, чтобы ваш настольный Linux подписался на данные «temp» в этом проекте, придется запустить

```
$ mosquitto_sub -d -h 192.168.0.13 -t temp
```

Параметры следующие:

-d Режим отладки, он показывает все ошибки и все шаги, предпринятые клиентом MQTT.

-h Указывает IP-адрес Посредника.

-t Тема, на которую мы хотим подписаться.

Если вы заинтересовались MQTT, можете почитать об этом дополнительно, перейдя по ссылке <http://mqtt.org>.

```
hall = temp
client.publish("temp", hall)
print(hall)
time.sleep(10)
```

Завершив код, не забудьте сохранить свою работу, и нажмите на Run > Run Module [Запустить > Запустить Модуль]. Теперь данные будут отправлены Посреднику — MQTT и напечатаны в оболочке Python, для показа, что код работает.

► Подписчик [Subscriber]

Наша последняя часть — Подписчик, устройство(а), которые слушают определенную тему. Подписчик — это другой Pi Zero W с подключенной платой Blink LED. Мы используем Blink для отображения температуры у нас в прихожей с помощью восьми светодиодов на плате.

Начнем настройку Pi-Подписчика, открыв терминал и установив библиотеку MQTT для Python:

```
$ sudo pip3 install paho-mqtt
```

Затем запустим скрипт установки Blink, созданный командой Pimoroni, который автоматизирует установку библиотеки и примеров для иллюстрации возможностей плат. В терминале введите

```
$ curl -sS https://get.pimoroni.com/blink | bash
```

Следуйте за скриптом, и через некоторое время на вашем Подписчике Pi установятся библиотека Blink и примеры.

А теперь приступим к коду Python 3 для этого проекта. Создайте новый файл и назовите его **MQTT-Subscriber.py**. При импорте библиотек для Подписчика мы начинаем с библиотеки MQTT, а затем импортируем ряд функций из библиотеки Blink, а именно настройку цвета пикселя, яркости, отображение изменений, очистка пикселей и изменение цвета каждого пикселя.

```
import paho.mqtt.client as mqtt
from blinkt import set_pixel, set_brightness, show, clear, set_all
```

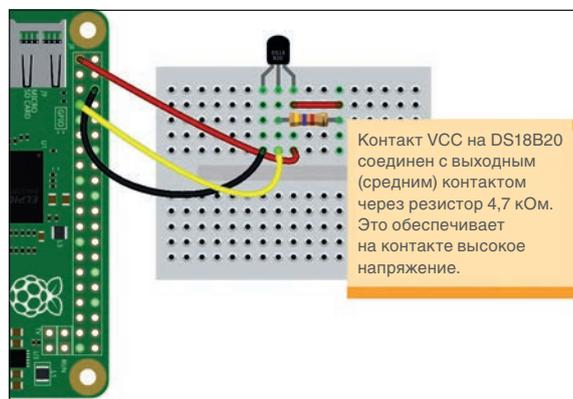
Установите яркость для Blink LED в 0.1, и в следующей строке мы обеспечим отключение всех светодиодов.

```
set_brightness(0.1)
clear()
```

Создайте функцию, которая будет работать с подключением Подписчика к Посреднику MQTT. Функции надо передать аргументы — IP-адрес Посредника, порт подключения и время ожидания соединения. Функция также будет печатать полученный в результате код в зависимости от того, смогли ли мы успешно подключиться к Посреднику. Затем она подпишется на тему «temp».

```
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("temp")
```

Наша следующая функция обрабатывает сообщения, отправленные Подписчику через MQTT. Она создает переменную с именем



► Мы использовали оконечный блок для подключения нашего датчика к Pi Zero W. А также подключили резистор между красным и желтым проводом, прикрепленным к оконечному блоку.

message, которая конвертирует сообщение в строку. Затем мы нарезаем переменную, удалив ненужную информацию, и сообщение конвертируется в float, после чего печатается в оболочке Python для целей отладки.

```
def on_message(client, userdata, msg):
    message = str(msg.payload)
    message = message[2:(len(message)-1)]
    message = float(message)
    print(message)
```

Используйте условный тест if, else if для проверки содержимого переменной message. Первый тест нужен для просмотра, что температура ниже 10°C и выше или равна 0°C, затем печатается сообщение об отладке в оболочке. Затем мы устанавливаем все LED в голубой, используя значение RGB (0,0,255), чтобы включалась только голубая часть LED. Затем надо вызвать функцию show(), чтобы увидеть изменения.

```
if message < 10.0 and message >= 0.0:
    print("В прихожей холодно")
    clear()
    set_all(0,0,255)
    show()
```

Создайте объект с именем client, который сокращает ссылку на библиотеку MQTT. При подключении запустите функцию on_connect и затем функцию on_message. Затем мы подключимся к IP-адресу нашего Посредника MQTT, указав порт и время ожидания подключения. И, наконец, дадим инструкцию клиенту MQTT постоянно следить за кодом, ожидая прибытия сообщения. Сохраните код и нажмите на Run > Run Module [Запустить > Запустить Модуль]. Вы должны увидеть, как LED на Blink реагируют на отправку данных через MQTT.

Поздравляем, теперь вы можете отправлять данные и работать с ними по сети с помощью MQTT. **LXF**





Глас народа

Политические взгляды Джонни Бидвелла уже закатали его в самые разные списки. Пообщавшись с **Джеймсом Смитом**, он уверовал, что будущее — за открытой демократией.



Джеймс Смит [James Smith] — глава лабораторий в Open Data Institute. Как и многие британцы, Джеймс (в Твиттере — @floppy) недоволен политической ситуацией в стране. Однако, в отличие от многих, он решил приложить свои умения в программировании к выражению своей позиции, сформировав свою политическую партию, Something New [Нечто новое]. Он выходил на выборы 2010-го, и хотя опыт оказался неудачным,

О ПАРТИИ 'SOMETHING NEW'

Мы — партия, не желающая существовать. Нас не должно быть.

он продолжает развивать идею создания манифеста партии Open Source, ключевого понятия в идеологии Something New. Мы встретились с ним после его выступления на OSCON Europe, чтобы обсудить его опыт. Его призыв к действию, под заголовком «Время сделать шаг», можно прочитать на <https://floppy.org.uk/blog/2013/07/05/its-time-to-step-up>.

Linux Format: До вылазки в это логово хищников, называемое политикой, вы были инженером-программистом — чем именно вы занимались?

Джеймс Смит: Я и сейчас инженер-программист — пока еще не совсем перешел на темную сторону; но да, я много чем занимался: биометрикой, какое-то время авиатренажерами, в последнее время — web-технологиями. Несколько лет проработал в стартапах, занимающихся изучением изменения климата, потом — в тех, что имеют дело с открытыми данными, отслеживаемостью и прозрачностью информации в них. Всегда пытался сделать мир лучше.

LXF: Как вы пришли в Open Source?

ДС: Ох, пожалуй, это было еще когда я писал кандидатскую, в 1999-м. Мы рабо-

тали с открытым кодом и боролись за то, чтобы весь остальной код тоже стал открытым. С тех пор с этим связана практически вся моя карьера.

LXF: Вы упомянули, что работали над повышением политической прозрачности, не могли бы вы рассказать об этом побольше? »

ДС: Да, это был проект под названием Save Parliament [Спасти парламент] — такая вот простая и достижимая цель. В 2006 г. был принят Законопроект о реформе законодательной и нормативно-правовой базы [Legislative and Regulatory Reform Bill] — звучит как нечто очень и очень скучное. Но там был пункт, где говорилось, что министры могут изменять законодательство через «постановления правительства» — кажется, так. По сути, это означало разрешение вносить поправки в законодательство, не пропуская их через парламент. Если вы верите, что никому из нынешнего или будущих правительств не захочется использовать это в своих целях — прекрасно, но такими вещами легко злоупотребить, особенно при неудачной формулировке. А формулировка была ужасной. Не помню, как я об этом узнал: об этом законопроекте писали всего несколько блоггеров, плюс пара статей в газетах. Ну и я тоже написал в свой блог, и потом мы, объединившись, совместно работали над кампанией, которую я в итоге координировал, так уж вышло. Мы поговорили с журналистами и опубликовали еще кое-какие материалы, подняли несколько вопросов и развернули нечто вроде предвыборной кампании. Мы обратились к депутатам, и текст законопроекта был изменен, что хорошо. Законопроект действительно прошел, но опасные фрагменты были удалены. Здорово чувствовать, что ты можешь на что-то повлиять, так что опыт был интересный и полезный.

LXF: Я думаю, многие мои ровесники, и, вероятно, большинство тех, кто помоложе, решили быть вне политики. Лично я пока нет, просто потому, что мой отчим — один из немногих независимых и, возможно, куда более немногих трудящихся членов городского совета в Великобритании. Он занимает этот пост много лет, потому что он абсолютно предан людям, для которых

он работает, и если напротив дома миссис Эклс вдруг перегорит фонарь, он живенько побеспокоит управляющие службы и не отстанет от них, пока фонарь не исправят.

ДС: Хорошо, что есть еще такие преданные люди, которые не забывают о тех, для кого они работают, — это обнадеживает. Но, безусловно, на более высоких уровнях политики она, к сожалению, рассматривается скорее как власть, а не как служение. Вы упомянули, что молодые люди уходят от политики; с одной стороны, это так, но с другой — абсолютно неверно. Да, политики могут думать, что молодые люди не хотят участвовать: они не голосуют, они не доводят свое мнение до этих институтов. Но это вовсе не означает, что молодым людям, а говоря о молодежи, я имею в виду свой возраст (40) и ниже, всё равно: просто они выросли в другом мире. Мир изменился, а институты — нет. Мы можем гораздо чаще и яснее выражать свои мнения, свои чувства по тем вопросам, которые нас волнуют, и обсуждать их между собой. Мы делаем это, но не так, чтобы существующие институты нас услышали. Так что, по-моему, дело не в том, что молодежи всё равно: просто существует разрыв между тем, как устроены мы, поколение Интернета, и старыми институтами, которые еще не адаптировались.

LXF: Конечно, бурно реагировать на политику легко — лично мне достаточно просто посмотреть новости; но столь же активно пожелать участвовать в ней гораздо труднее.

ДС: Но это и станет частью перехода к новому типу политики и новому типу демократии, где мы найдем способы, или они появятся сами, чтобы расширить возможности участия в политике. Как сделать так, чтобы твои взгляды и настроения были услышаны и поняты без полного погружения в это, не вступая в какую-либо политическую партию или

нечто подобное? Как помочь людям влиять на политику с помощью более легких методов, чем те, к которым мы привыкли сегодня?

LXF: В вашем выступлении был такой хороший тезис: «политика — самое большое препятствие». Значит ли это, что создав свою политическую партию, вы пытаетесь изменить что-то изнутри?

ДС: Идея состоит в том, что вы должны начать с того, где находитесь. Я хотел бы быть в системе, где не нужны политические партии. Я хотел бы быть в системе, где есть представители, потому что идея делегативной демократии и динамичного обмена кажется мне очень интересной. Но сейчас такого нет, сейчас даже не та ситуация, когда у людей спрашивают их мнение, имитируя, что они могут на что-то повлиять. Люди не верят, что система хочет их слышать. Вот почему я думаю, что идет смена поколений и что понадобится много времени, чтобы этот менталитет сменился готовностью выражать свое мнение и делать это регулярно. Пока мы настолько далеки от этого идеала, что потребуются очень долгое время на это.

LXF: Идея переворота старых институтов с ног на голову отчасти возвращает нас к тому, что Ричард Столлмен [Richard Stallman] сделал с законом об авторском праве и GPL, когда, придумав copyleft, он совершил своего рода революцию. Закон об авторском праве был скорее об ограничении прав людей и обеспечения закрытости, а с GPL всё стало наоборот, и все разработчики должны были открывать исходники.

ДС: Именно, вопрос в том, как взять существующую систему и направить ее в том направлении, в котором она, по вашему мнению, должна развиваться. Потому что есть куча людей, придерживающихся позиций прямой демократии, которые говорят: «Если вы выберете меня, то на следующий день после выборов я проголосую в парламенте именно так, как вы этого хотите, я во всё буду опираться на вас». Это действительно благородная идея, но не думаю, что мы к этому готовы. То есть, поймите меня правильно, я хотел бы, чтобы это было так, я верю в лучшее в людях. По-моему, если людям доверять, то они, в целом, оправдают это доверие. Но я не думаю, что мы сможем сделать это за одну ночь, потому что, естественно, есть медиа-культура, есть много проблем, с которыми вам придется иметь дело, прежде чем удастся к этому прийти. Я имею в виду, что результат последнего референдума [Brexit] — это всего лишь пример того, как опрос общественного мнения перерастает в шокирующее проявление демократии. На самом деле, последние два референдума были ужасными (как по альтернативному голосованию, так и по евро), и не только из-за результатов, но просто потому, как это делалось. Так что понадобится много времени, чтобы всё наладить, и пока мы находимся на пути к этому, мы являемся партией, которая не желает существовать. Мы охотно бы вымерли, в будущем нас не должно быть.

LXF: Это моя вторая конференция OSCON, мои боссы нечасто выпускают меня из клетки, но в тех



редких случаях, когда мне доводится пообщаться с живыми людьми, я понимаю, что Open Source — это уже гораздо больше, чем просто возня со скриптами POSIX или дирижирование контейнерами Docker. В каком-то смысле ваши усилия выражают позицию наиболее активных сторонников идеологии Open Source: сформировать политическую партию за Open Source.

ДС: Я считаю, главное, что принесло движение Open Source — это сотрудничество. Это позволяет людям работать вместе, облегчая доступ к сложным вещам, чтобы каждый мог внести свой личный вклад.

LXF: Как же именно инструменты Open Source помогают разжечь политическую революцию?

ДС: Сам манифест — просто куча текстовых файлов, хранящихся в *Git*. Все они выполнены на языке разметки, которая достаточно интуитивно понятна, но возможности изменения со стороны пользователей мы ограничиваем. Нельзя позволить, чтобы всё это начало мигать, или шрифт сменился на Comic Sans, или еще какой-нибудь ужас в том же духе. Важна идея разделения формы и содержания. Основные функции *Git*, которые мы используем и которые сегодня обязательны для прочих распределенных систем контроля версий, — это простое слияние и непрерывное ветвление. Раньше считалось, что ветвление — «зло», что ветки в проекте — это скандал. Теперь ветки добавляются ежедневно, став просто частью рабочего процесса, и я думаю, это результат и развития распределенного управления версиями, и тех новшеств, которые GitHub, хоть это и закрытый код, принес в мир Open Source. Ведь даже работать над FOSS-проектами с использованием SourceForge раньше было намного сложнее — барьеры вхождения были намного выше. А теперь есть абсолютно бесплатная ветка, куда можно внести изменения, добавить что-то от себя, отправить запрос на включение, и не надо никого уговаривать, это очень действительно. Вместе со стандартными функциями управления версиями, такими как возможность отката, изменение журналов, это очень здорово. Как и то, что в партийном манифесте можно запустить `git blame`.

О МАНИФЕСТЕ OPEN SOURCE

Главное, что принесло движение Open Source — это сотрудничество.

LXF: И я слышал, у вас тоже есть робот подсчета голосов. Как это работает?

ДС: Довольно просто: это было маленькое приложение *Sinatra*, которое теперь выросло в приложение *Rails*. Я программирую на Ruby, а когда у вас Ruby, всё выглядит как написание приложения *Rails*. Еще один большой плюс GitHub в том, что вокруг него существует обширная экосистема инструментов, и можно использовать ее как платформу. Приложение перехватывает оповещения `webhooks` на GitHub и получает уведомление



о любых изменениях, оно знает, когда возникают проблемы, и у него есть своя база данных. Кроме того, оно пробегает комментарии, находя символы смайликов «нравится» или «не нравится». Недавно API начал выдавать смайлики, а не текстовую версию, и мне пришлось изменить код — в нем теперь тоже есть смайлики, хотя у меня от этого чуть крышу не снесло.

LXF: Даже простой текст подвергся веяниям нового тысячелетия.

ДС: Точно! Ну да, это открытый код, следующий простым правилам и хорошо протестированный. В нем можно протестировать правила голосования и прочее, а потом всё вновь загружается на GitHub API через статус сборки. Опять же, вся платформа-то уже есть, надо только добавить кое-какие связующие элементы, которые будут делать то, что вам требуется. Все инструменты, которыми мы располагаем, это умеют. Работали мы в Heroku — это открытая инфраструктура. По сравнению с тем, что было двадцать лет назад, когда я начинал, эти инструменты просто феноменальны.

LXF: Я пишу много вводных статей по Python, и да, вы можете обойтись теми же привычными алгоритмами, что и при изучении языков программирования. Но можно и просто взять каких-нибудь 20-30 строк кода и несколько готовых доступных

модулей, которые подключаются к тому-сему. Графике, live-данным, обработке звука.

ДС: Да, это потрясающе. Это немного не по теме, но один мой знакомый работал в команде создания искусственного интеллекта Watson в IBM. Половину этого времени он занимался программированием со своими детьми, работая с публичным API Watson. То есть восьмилетка и десятилетка работали с крупнейшей системой ИИ. Технология ускоряется, она становится всё быстрее, у нас есть системы, которые мы уже не понимаем, мы не знаем, как будет выглядеть мир через пять лет, не говоря уже о столетии. Вот почему я считаю, что нам нужна цель, нечто вроде «куда мы идем?». Я вдруг понял, что забыл объяснить один из моих слайдов, о том, в каком фильме или ТВ-шоу о будущем вы бы хотели оказаться? *Бегущий по лезвию* [Blade Runner]? *Звездный путь* [Star Trek]? Мне всегда казалось, что мы, скорее всего, попадаем в *Дети людей* [Children of Men], это кажется всё более вероятным. Но к чему мы стремимся как биологический вид? И я думаю, что возможности наших правительств — поскольку это они имеют дело с эволюцией общества — в этом плане очень ограничены.

LXF: Я думаю, мы сходимся на том, что политическая ситуация сейчас довольно напряженная, особенно в последние два-три месяца [после результатов Брексита]. В своем докладе вы упомянули, что мы всё еще «находимся в состоянии промышленной революции». Это странно, потому что мы выросли из промышленной революции как таковой; мы больше не являемся мастерской мира, но цепляемся за эту политику, потому что

не нашли ей подходящей замены. В качестве примера того, как может выглядеть эта замена, вы привели движение Podemos. Не могли бы вы немного рассказать об этом?

ДС: Podemos — это движение, возникшее в Испании. Ход их мыслей не совсем совпадает с нашим, они скорее ближе Оссигу, 99%. Но оно очень быстро набрало популярность, как и партия Five Star в Италии, которой удалось активно стартовать, впечатлить электорат и пробиться. А теперь в Барселоне мэр из Podemos, а посмотрите на Pirate Party в Исландии — кто бы мог подумать?

LXF: Да, похоже, что импульс определенно копится, но трудно понять, в каком направлении этот импульс пойдет.

ДС: Я думаю, что старая система начинает медленно, но верно давать трещины. В Великобритании мы, похоже, погрязли во внутренней борьбе: особенно печально наблюдать, как официальная оппозиция Ее Величества со дня после голосования по референдуму просто отошла в сторону и ничего не предпринимает. Это так досадно... и зачем?! Так тратить энергию и совершенно впустую. Я не принимаю ничью сторону, но зачем всё это затевалось?

Всё это крутится вокруг внутренних распрей и политических интриг, а не реальных проблем. Этим референдумом правительство наложило на себя ответственность, и многое из заявленного они пока что не выполняют. И виной всему, я думаю, как раз все эти институты, и, как я уже сказал, я не уверен, что они могут измениться. Это как в книге *Дилемма инноватора* [Innovator's Dilemma], где говорится о цикле разрушения малых и крупных компаний — почему большие компании не могут приспособиться, и т.п.

Одна из причин, почему крупные компании не могут внедрять инновации, и я бы сказал то же самое в отношении любого крупного учреждения, заключается в том, что у них есть постоянная клиентская база, желающая, чтобы всё оставалось как есть. От этого очень, очень трудно уйти. То же наблюдается и в менее популярных партиях, например, в Партии зеленых. У них очень специфическое представление об экологии, очень консервативное. Я считаю себя защитником окружающей среды, но не разделяю взгляды Партии зеленых. Идея о том, что нам надо забыть о ядерной энергии, раз мы пытаемся справиться с изменением климата, мне совершенно непонятна.

Думаю, нам следует быть чуть более прагматичными. Не способные к изменению системы строятся на подобных аксиомах, потому что у них есть свои ценности, которые, как мне кажется, должны быть подорваны — не столько потому, что это всё изменит, но потому, что это спровоцирует более глобальное изменение. Мне всё равно, кто тут выйдет победителем или кого им назовут, но я знаю, что этот переход должен произойти. Поскольку если и через 50 лет система останется той же, то я не думаю, что она будет работать — она и сейчас не очень работоспособна. Что-то должно измениться, и я не знаю, что именно, но здорово пытаться и быть частью этого.

LXF: А как другим людям в этом поучаствовать? Кое-что, я уверен, будет очень созвучно нашим читателям. Кто нужен Something New?

ДС: Да все. Нужны люди, которые умеют создавать сообщества. Нужен дизайн, UX, разработчики. Нам нужны люди из всех сфер, чем разнообразнее — тем лучше. Пока что большинство участников, с кем я встречался, были довольно состоятельными белыми парнями. Есть и другие, с кем я не знаком, знаю только их имена, но я серьезно сомневаюсь, что они существенно изменят картину. Это не тот подход, чтобы построить инклюзивную платформу.

LXF: Это огромная проблема во всей технологической индустрии, не так ли?

ДС: Это глобальная проблема. Сам я, будучи белым мужчиной средних лет, не думаю, что смогу с ней справиться, поэтому очень хочу услышать от других, как нам это исправить. Но я остро осознаю, что, говоря это, я сам являюсь частью проблемы. Поэтому мы пытаемся вовлечь как можно больше людей, чтобы создать импульс и продолжить движение, которое превратится неизвестно во что. **LXF**



softline®

Cloud Software Hardware Services

20+

Years in IT

IT-архитектура вашего бизнеса



корпоративные
мобильные решения
на платформе Microsoft

SaaS

CLOUD

PROJECTS

гибридные облака

SOLUTIONS

remote workplace

SAM

поставка аппаратных
решений
юридическая
поддержка

education

Mirapolis

SECURITY
учебный
центр

разработка IT-стратегий

hardware
ERP&CRM
CAD
CAM
CAE

VIRTUALIZATION

web-разработка legal support

**защита
персональных данных**

SOFTWARE

публичные облака

САПР и ГИС

LINUX

**программное
обеспечение**

BI & BigData

SERVICES

WinSpace

**информационная
безопасность**

решения Microsoft

data leak
prevention

PaaS

разработка информационных систем
**корпоративный портал
DeskWork**

open
source

техническая поддержка

search

flexible

CONSULTING

PLATFORM

IaaS

частные
облака

protection

hosting

technical support

16+

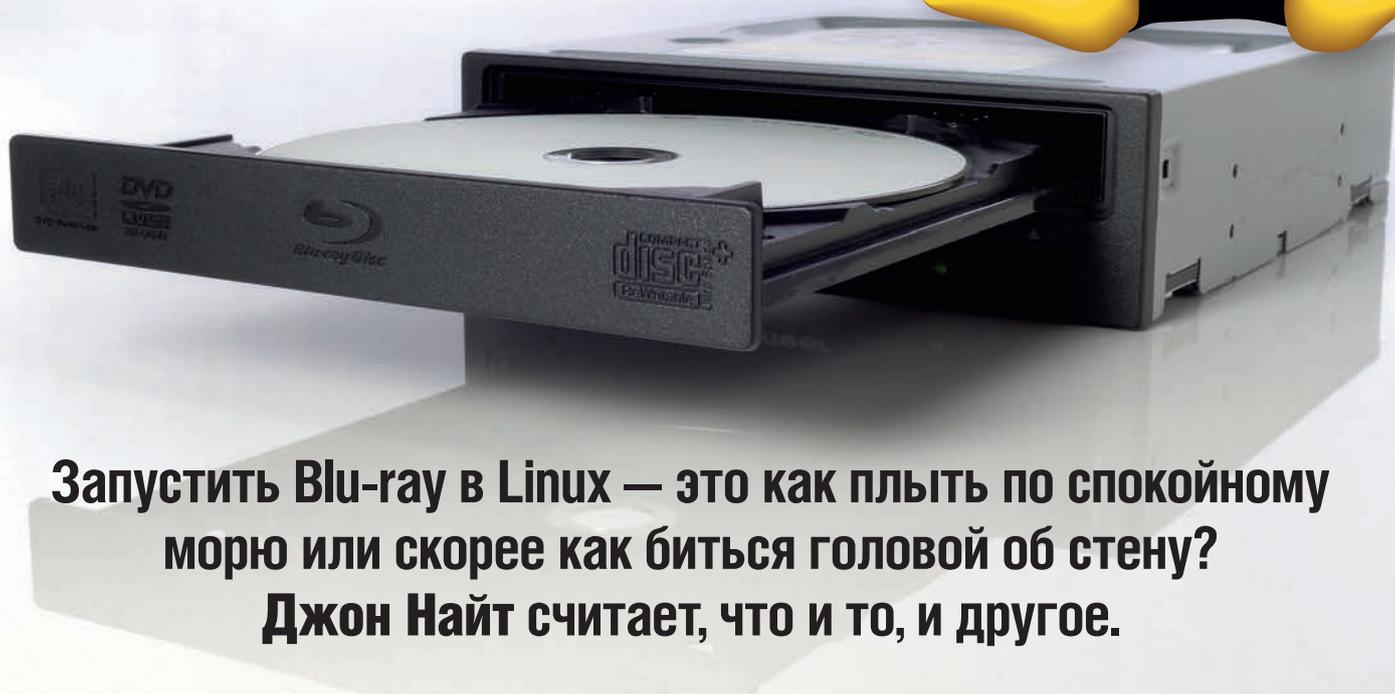
8 (800) 232-00-23

+7 (495) 232-00-23

www.softline.ru

info@softline.ru

Blu-ray в Linux



Запустить Blu-ray в Linux — это как плыть по спокойному морю или скорее как биться головой об стену? Джон Найд считает, что и то, и другое.

Любой пользователь Linux, который пытался запустить диск Blu-ray, делал малоприятное открытие, что сразу он воспроизводиться не будет и что для этого придется побегать. Быстрый поиск в Google показывает разные степени мрачности, и в зависимости от того, когда было написано сообщение на сайте, вы получите разную информацию о том, как воспроизводить Blu-ray и насколько вообще полезен Linux/Blu-ray в данный момент.

Проблема заключается в защите от копирования и в юридических вопросах. Хотя Linux

более чем способен воспроизводить Blu-ray с отличной производительностью (велики шансы, что ваш плеер Blu-ray на самом деле работает на ядре Linux), с самого своего появления диски Blu-ray находятся под неусыпным кон-

Велики шансы, что ваш плеер Blu-ray на самом деле работает на ядре Linux.

тролем драконовской формы управления цифровыми правами. И поверх этого накладываются различия в законодательстве разных стран по поводу того, что можно и чего нельзя делать

с мультимедиа с цифровой защитой, что создает для разработчиков дополнительную головную боль. А для пушечного упрощения, есть кровные интересы крупнейших корпораций, включая Microsoft, которые в свое время ничуть не горели желанием предоставить пользователям Linux легкий доступ к Blu-ray.

Так что же, ситуация для пользователей Linux — полный мрак? Как нам воспроизводить Blu-ray и почему это настолько сложно вообще? Давайте посмотрим, как в 2017 г. выглядит Linux/Blu-ray, и попробуем заставить его работать.

Проект: www.labdv.com/aacs

Если вы столь же ленивы, как мы, то вы посмотрите на инструкции в этом URL и подумаете: «слишком сложно, незачем и пробовать». Однако не спешите — всё проще, чем кажется. Вам нужно будет установить некоторые библиотеки, скачать файл и куда-нибудь его скопировать. Вот и всё. Две минуты — и готово.

Сначала откройте свой менеджер пакетов и установите пакеты *libbluray* и *libaacs* (вероятно, стоит также установить пакеты *-dev*, пока вы там). Теперь скачайте файл ключей Blu-ray с указанного URL (имя файла **KEYDB.cfg**), и скопируйте его в папку `~/config/aacs`.

Для всех начинающих пользователей: все файлы или папки, имена которых начинаются с точки, являются скрытыми. И если вы хотите работать в графическом менеджере файлов, включите Show Hidden Files [Отображать Скрытые Файлы] и откройте папку **.config**, скрытую в вашей директории **Home**. Если папки **aacs** не существует, создайте ее и скопируйте туда файл ключей.

Для ворчливых ветеранов командной строки, просто введите эти две команды:

```
$ mkdir -p ~/config/aacs/
$ cd ~/config/aacs/ && wget http://vlc-bluray.whoknowsmy.name/files/KEYDB.cfg
```

Теперь давайте-ка разберемся сразу: есть различные шансы, что ваш диск не будет работать. Не паникуйте, потому что мы выясним, что делать,

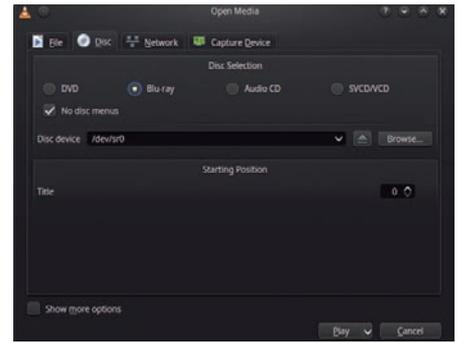
но из десяти дисков, которые мы испробовали, три не работали. Причина в проблемах с шифрованием, которым посвящена вторая часть нашей статьи; а сейчас соберите побольше дисков Blu-ray, чтобы увеличить свои шансы найти работающие диски.

В отличие от DVD, вам сначала придется подмонтировать диск. Воспользуйтесь менеджером устройств своего рабочего стола — здесь вам не придется вводить затейливых команд.

VideoLAN — отличный стартовый пункт, поскольку это самый простой способ выяснить, будет ли диск работать, не обращаясь к терминалу. Нажмите на диалоговое окно Open Disc [Открыть Диск] и выберите Blu-ray. Вы увидите, что отмечено No disc menus [Без меню диска] — пока оставьте всё как есть, мы вернемся к этому позднее. Теперь нажмите на Play, и если вам повезет, ваш Blu-ray запустится. Если нет, переходите к следующему диску.

Поздравляем, если ваш фильм проигрывается без инцидентов — к сожалению, велики шансы, что у вас есть проблемы с воспроизведением, например, подтормаживание и щелчки. Тем не менее, вы, возможно, сумеете их обойти, повозившись с настройками — например, снизив качество (что, вообще говоря, лишает Blu-ray смысла) или настроив все видеодрайверы на “VA API”.

Теперь давайте попробуем *MPlayer*. Наш любимый GUI, *SMPlayer*, не имеет диалогового окна для воспроизведения Blu-ray, так что вам, возможно,



➤ Меню Blu-ray будут отключены во всей системе по умолчанию, но это же истая роскошь — не видеть бесконечных экранов предупреждений!

придется обратиться к терминалу, так же как нам. Но не паникуйте, поскольку это очень просто. Предположим, имя устройства вашего диска “sr0”; введите следующую команду — в ином случае просто замените имя устройства вашего диска:

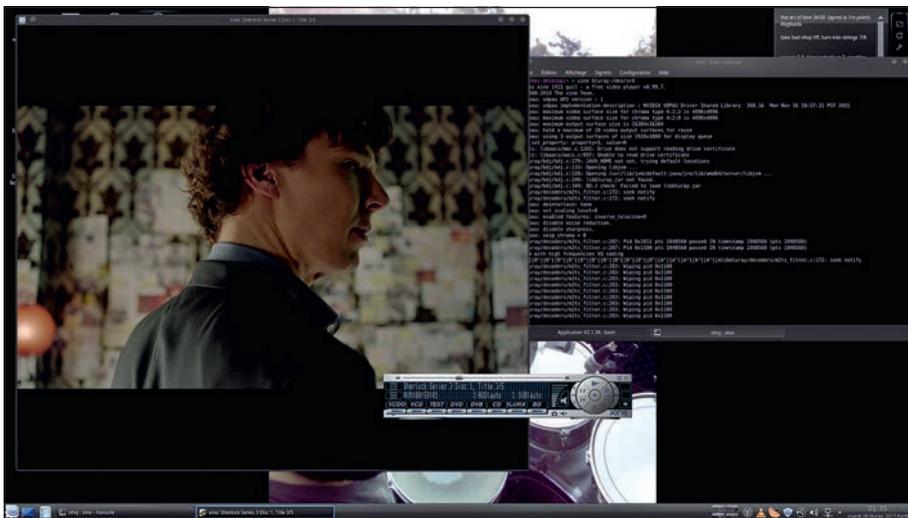
```
$ mplayer br:///dev/sr0
```

И да, нужны четыре слэша — не спрашивайте, почему.

На нашем компьютере воспроизведение было куда стабильнее, чем в *VideoLAN*, и не было никаких пропусков, выпадения кадров или графических дефектов. Но — и это немалое «но» — попытки осуществить поиск по видео приводили к сбою *MPlayer*, тогда как *VideoLAN* прекрасно с этим справлялся. Более того, если вы не отредактировали свой файл настройки, *MPlayer* по умолчанию включает режим стерео; а значит, если вы хотите получить объемное звучание, вам придется повозиться с весьма корявыми параметрами командной строки (что отнюдь не весело).

Ни один из этих плееров нас не устроил, и мы с удивлением обнаружили, что спасителем оказался наш давно забытый друг *xine*. Если память нам не изменяет, GUI *xine* ведет начало с 1990-х, но не стоит сбрасывать его со счетов, потому что: (а) у нас он работал куда лучше, чем *VideoLAN* или *MPlayer*, и (б) с ним проще настроить ваш драйвер и звуковые опции (колонки, цифровой или аналоговый звук с помощью драйвера Nvidia VDPAU и т.д.).

Хотя в *xine* имеется кнопка Blu-ray, у нас она не сработала, поэтому вам, возможно, придется использовать терминал. Но опять же — не пугайтесь: это даже проще, чем *MPlayer*! Просто введите команду и наслаждайтесь результатами: ➤



➤ VLC пропускал кадры и пощелкивал, MPlayer выдавал сбой при каждой попытке поиска, а старый добрый xine работал идеально.

Осторожно: Старые сайты!

Берегитесь старой web-страницы, которая может оказаться первой в результатах вашего поиска в Google, URL: <http://vlc-bluray.whoknowsmy.name>. На ней предлагается старый файл ключей начала 2016 г.: не используйте его! Не то коэффициент результативности вашего Blu-ray упадет с семи из десяти до каких-нибудь четырех из десяти — ошибка, обнаружение которой далось нам нелегко! Эта ссылка может быть по-прежнему включена в ряд руководств и справочников, и если вы сами

когда-то писали документацию, пожалуйста, обновите ссылку.

Правильная ссылка: www.labdv.com/aacs. Посетив эту страницу, вы, вероятно, обзаведетесь файлом базы данных ключей за последние несколько дней, а не старым файлом 2016 г. Кроме того, включены ссылки на дальнейшие библиотеки и полезные программы, а также на добавленную документацию, плюс крайне важная ссылка на загрузку ключей. Обратите внимание, что ряд

руководств и программ размещены на форумах doot9, а не на центральной web-странице, что делает процесс несколько анархичным: некоторые ссылки уже не работают, а некоторым руководствам не помешает обновление (к несчастью, они как раз относятся к критически важным областям, если вы — пользователь Linux, пытающийся сгенерировать ключ Blu-ray). Из-за этого мы несколько недель бегали кругами, так что будем надеяться, что их старая информация скоро обновится.

А это законно?

В Великобритании частное копирование защищенных авторским правом материалов является незаконным, однако данная статья посвящена проблеме правомерного доступа к законно приобретенным Blu-ray. К сожалению, в Linux для этого требуются цифровые уловки — что несколько лет является темой горячих дебатов и постоянных столкновений по поводу Прав на добросовестное использование.

В США *Digital Millennium Copyright Act* (DMCA) объявляет незаконным дешифрование защищенного контента (даже при добросовестном приобретении), и в судах США до сих пор идут сражения

DMCA против Добросовестного использования, так что вам придется смириться с тем, что в США эта практика остается нелегальной.

В теории законодательство Великобритании весьма драконовское, но на практике — либеральное, и годами подвергается обсуждению, поскольку интересы владельцев авторских прав в UK постоянно вступают в противоречие с реальностью. Личные копии законны, если вы живете в Нидерландах, Испании или Австралии, но не распространяйте эти материалы. Эта область довольно мутная, потому что все эти законы в первую очередь направлены

на ограничение предоставления общего доступа к контенту, но не к ключам.

И чтобы сделать почву еще более зыбкой, инструменты обхода могут быть законны, если не нарушают патент, а вот производство и распространение будет незаконными. Юридические исключения для обхода могут существовать для журналистики [Ред.: — Это мы, мы!], критики или комментариев, а также для некоммерческого видео.

Так законно ли это? Мы не юристы, и не в состоянии ответить на этот вопрос. Вам лучше обратиться к знающему консультанту.

```
$ xine bluray:/dev/sr0
```

В конечном итоге, если *VideoLAN* работает без проблем с воспроизведением, выбирайте его: вы можете осуществлять поиск без сбоев и даже использовать меню, если у вас установлены экспериментальные библиотеки. Но самое главное — вы можете открыть диск, не обращаясь к терминалу. Тем не менее, вышеупомянутые проблемы с Blu-ray являются весьма обычными, хотя команда *VLC* обещает значительно улучшить это в версии 3.0. А пока старый добрый *xine* будет вашим лучшим выбором, если вы не против ввода простых команд в терминале.

MakeMKV

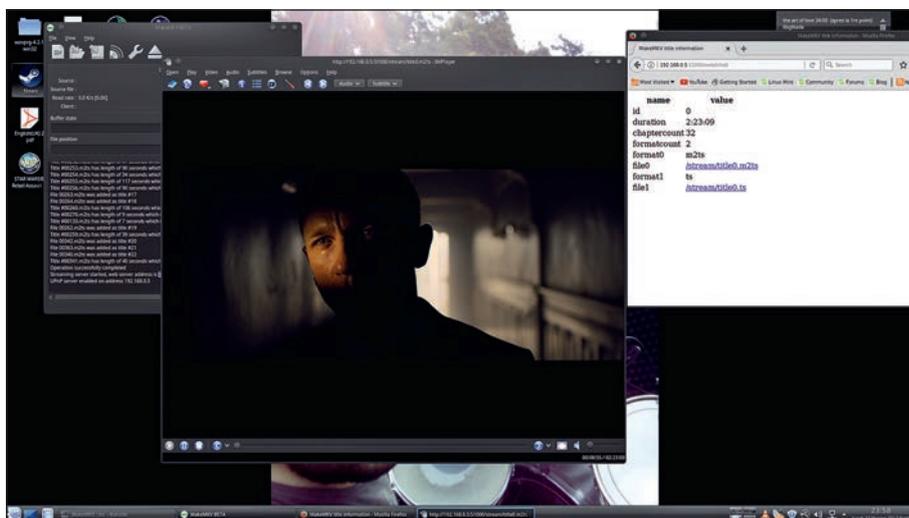
Так почему же не работают некоторые ваши диски? Потому что для просмотра зашифрованного Blu-ray — а таковыми является большинство из них — нужен ключ, из тех ключей, которые были в файле, скачанном вами ранее. Если ключа там нет, нет и доступа. Огорчены? Что ж, выход есть... на время.

MakeMKV — это в первую очередь программа нарезки для дешифрования Blu-ray, чтобы преобразовать содержимое диска в видео Matroska, но также у нее есть чудесная способность потокового вещания фильмов на локальный порт, которое потом может запускаться почти в любом плеере. А в чем подвох? Прежде всего, это условно-бесплатная программа (мы не сообразили, что ведь это 1992 г.), и у нее имеется 30-дневный пробный период, после которого вам предлагается за нее платить. Тем не менее, это чертовски хорошее приложение, и оно достойно вашего внимания.

Перейдя на bit.ly/1hJcdn, вы найдете там ветку форума с версией для Linux и инструкциями по установке. Должны с сожалением отметить, что у нас



► Воспроизведение «Полночи в Париже» в *xine* было таким же прекрасным, как и сам фильм.



► *MakeMKV* + браузер + медиа плеер = воспроизведение любого Blu-ray, по вашему желанию... в течение 30 дней.

нет места на рассказ об установке, но мы можем сообщить вам, что она немного затруднена наличием двух tar-архивов; однако они не сложнее в установке, чем обычный tar-архив исходника. К счастью, хозяева были достаточно любезны, чтобы сообщить названия пакетов для необходимых библиотек и даже команду для производных Ubuntu.

Когда вы ведете потоковую трансляцию диска, вы не можете просто скопировать URL, предоставленный *MakeMKV*, и вставить его в свой плеер: ничего не произойдет. Сначала надо открыть URL в браузере, где потом и осуществлять навигацию по структуре диска. Выберите номер требуемого произведения, скопируйте местоположение ссылки из браузера и затем можете вставить нужный URL в свой любимый плеер.

Кстати о плеерах: нам показалось, что *VideoLAN* был глючным с постоянными сбоями видео, а вот *SMPlayer* продемонстрировал отличное воспроизведение. Обратите внимание, что если у вас нет звука, вам, возможно, придется изменить поток аудио в меню вашего плеера: аудиопотоки могут, например, не воспроизводить звук, если это прямой объемный сигнал, а ваш звук настроен на аналоговое стерео. Если у вас есть многоканальный усилитель мощности, выберите цифровой выход (AC3/DTS pass through S/PDIF в *SMPlayer*) в Sound

Options [Опциях Звука]. (Если у вас дешевая звуковая карта, то вам не повезло, извините.)

Злобное шифрование

Так почему же Blu-ray настолько сложно поддерживать, и почему мы не можем просто взломать его шифрование так, как мы обычно делали с DVD? Поясняет Жан-Батист [Jean-Baptiste] из команды *VideoLAN*: «Blu-ray довольно сложно поддерживать по двум основным причинам: DRM и Интерактивные Меню. Справиться с видеочастью Blu-ray несложно, поскольку это обычные кодеки, а вот пакеты очень сложны в обработке.

На самом деле, меню, навигация, бонусы и игры могут быть выполнены в простом формате, подобном меню DVD, или в среде исполнения Java под названием BD-J. Как обычно, открытые спецификации отсутствуют, и даже если бы они и были, многие диски их не придерживаются, поэтому приходится прилагать немалые усилия.

DRM — это, конечно, вторая большая проблема. В Blu-ray имеется несколько уровней DRM. Два из них, представляющие для нас интерес, это AAC3 и BD+. AAC3 — это документированная криптосистема, использующая ключи шифрования и сертификаты. В сообществе открытого кода имеется ее хорошая реализация, однако без ключей в ней

мало пользы. А ключи есть для каждого носителя, для каждой программы и для каждого диска, и они постоянно аннулируются.

Чтобы вы получили представление об этом: когда вы ставите новый Blu-ray, он может аннулировать сертификат вашего привода, и тогда на этом устройстве вы не сможете читать никаких Blu-ray до тех пор, пока не обновите прошивку! Даже те, которые вы уже воспроизвели [Ред.: — Хорошо быть собственником!]. Поэтому люди и делятся огромными файлами настройки, содержащими ключи для дисков на разных сайтах. Не очень радостно.

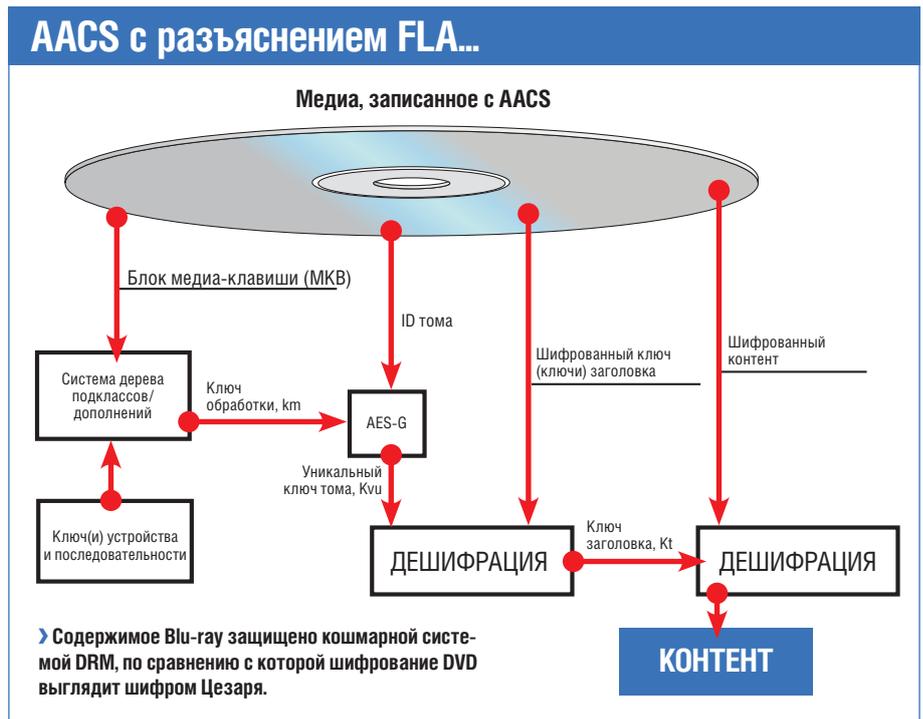
Вторая DRM — BD+, это полноценная виртуальная машина, исполняющая код DRM, который мог бы прилагаться к диску (мы не шутим). Ее очень сложно реализовать корректно, и она также интегрируется с уровнями Java и тоже имеет сертификаты. Хорошей реализации этой системы с открытым кодом нет».

Так как же простому человеку бороться с системой, чтобы получить законное право воспроизводить собственный диск? Ну, основной выход — предоставление ключей и отчетов об ошибках в меню, причем ключи являются наиболее востребованным ресурсом — хотя по юридическим причинам мы не можем этого поддерживать.

Будучи экспериментальной технологией, меню Blu-ray по умолчанию отключены. Согласно wiki Arch Linux, меню Blu-ray можно использовать, если вы установите *libbluray-git* вместо *libbluray*, так что вам, возможно, придется немного повозиться, если вы хотите, чтобы ваши меню работали, и поотсылать сообщения об ошибках.

А как насчет ключей? Здесь всё несколько усложняется. Если у вас по соседству есть Windows, то с вами всё хорошо: просто следуйте по ссылке, где вы сможете загрузить ключи, и читайте информацию о том, какие использовать программы и как их использовать. Зато Linux... у нас проблемы: *MakeMKV* больше не работает так, как говорят предлагаемые инструкции — они ссылаются на файлы и папки, которых больше нет — а дальнейшие программы и документация погребены где-то в лабиринтах бесконечных постов на форумах.

Проведя несколько недель в поисках вводов и опробовании разных методов, мы просто не смогли выбить из наших программ ключ. И хотя нам бы хотелось предложить вам простое руководство по поиску ключей, мы можем оказаться в неприятном положении — поэтому, с вашего разрешения, мы не будем вносить ясность в этот вопрос. В любом случае, следуйте по ссылкам и инструкциям



на форумах, и будем надеяться, вам повезет больше, чем нам.

Что дальше?

Возможно, вас огорчает такое положение дел, но семь из десяти дисков (по крайней мере, в нашем беглом тесте) — не такой уж плохой результат, разве нет? Плюс к тому, запуск без меню по умолчанию в своем роде удовольствие: вы не продираетесь сквозь бесконечные меню и экраны предупреждения, а за считанные секунды начинаете смотреть фильм. Рассмотрев, какова была ситуация с Linux Blu-ray несколько лет назад, мы можем подтвердить, что разработка продвинулась далеко, и будет развиваться семимильными шагами, если мы сможем поддерживать ее как сообщество.

Если отвлечься от ключей и меню, то от нас как от сообщества требуется старая нудная беготня и организация. Мы не хотим обижать разработчиков, но эти последние несколько недель были бесконечным блужданием по лабиринту старых и битых ссылок, устаревшей документации и кривых файлов начальной загрузки. Это можно понять. Людям за это не платят, а поддержка — весьма трудоемкий аспект любого программного проекта. Большинство программистов хотят просто

заниматься кодированием. Если вы хотите внести свой вклад, но не умеете программировать, не можете ли вы в выполнении этих скучных административных обязанностей и обеспечить обновленные инструкции, чтобы программисты могли уделять больше времени конкретным задачам?

Если нас читает сердитый деятель киноиндустрии, большинство из нас скорее всего отнюдь не собирается извлекать информацию с носителя, а просто хочет совершенно законно воспроизвести приобретенный диск, не переходя на другую ОС. Насколько мы знаем, коммерческих плееров Blu-ray для Linux нет, и хотя мы являемся сторонниками открытого кода, мы всё равно были бы заинтересованы в наличии проприетарной программы по разумной цене при отсутствии иных альтернатив — если вы не можете предложить нам ничего лучше.

Так где же мы находимся в плане завершенности проектов и усовершенствования Blu-ray? Предоставим финальное слово Жану-Батисту: «Что касается кода, он в основном написан. Что касается ключей, это будет долгая, бесконечная борьба».

Еще больше указателей на воспроизведение Blu-ray, намного больше, чем вы можете представить, вы найдете в превосходной Arch Linux Wiki: <https://wiki.archlinux.org/index.php/Blu-ray>. LXF

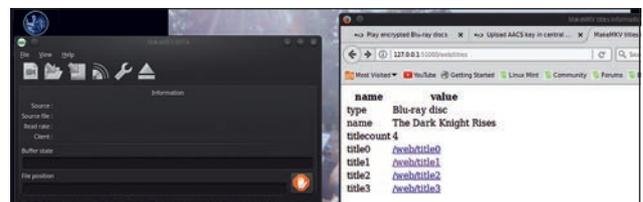
Потоковое воспроизведение с MakeMKV

Потоковое воспроизведение Blu-ray в *MakeMKV* сначала может запутать, потому что данный адрес не будет работать в вашем медиа-плеере. Не горюйте: это просто, как раз-два-три...

1) Скопируйте адрес в поле вывода текста *MakeMKV* (<http://localhost:51000>) и вставьте его в браузер.

2) Вы получите HTML-интерфейс для просмотра структуры диска: есть шанс, что фильм будет в "title1", и длительность файла должна помочь вам разобраться, какой файл чем является.

3) Скопируйте адрес требуемого фильма, и теперь вы сможете воспроизвести его практически в любом плеере.



Canonical- Кремень



Джонни Бидвелл не попал в Барселону на MWC 2017, но пообщался с Майком Беллом, исполнительным вице-президентом Canonical по устройствам и IoT, который там был.



Mobile World Congress 2017, который в этом году проходил с 27 февраля по 2 марта — это главная конференция для мобильной индустрии. Здесь

демонстрируются последние мобильные разработки от операторов телекоммуникаций, создателей Интернета Вещей (IoT) и инноваторов, и руководителей индустрии. Майк Белл [Mike Bell] дежурил у стенда Canonical и любезно согласился пообщаться с нами и поделиться своими впечатлениями. Ветеран индустрии, он был главным техническим директором Jaguar Land Rover, потом директором их программы Global Connected Car, а еще раньше — начальником отдела стратегии информационных систем и архитектуры [Information Systems Strategy & Architecture] в United Utilities. Основываясь на полученном опыте, он теперь возглавляет команду устройств и Интернета Вещей в Canonical, и мы были счастливы узнать, что же у них происходит.

Linux Format: Наш бюджет не предполагает возможности отправки меня в Барселону, поэтому — как прошел MWC 2017? Кто посещал ваш стенд, и чем они больше всего интересовались?

Майк Белл: С нашей точки зрения, это был наш лучший MWC — наш стенд посетило рекордное количество людей. Кроме того, для нас это было довольно напряженным шоу, уровень получаемого нами интереса был весьма разнообразным. Традиционно наш стенд посещали представители телекоммуникационных предприятий, но также и традиционные представители мира технологий, а также производители устройств — так что это была пестрая смесь. Большой интерес вызывали такие области, как умные города и робототехника, именно эти темы поднимались. Кроме того, была затронута и вызвала большой интерес демонстрация современных шлюзов от Dell и ряда других производителей. Мы также продемонстрировали оборудование Programmable Logic Controller (PLC),

на котором работала наша самая последняя ОС для IoT, Ubuntu Core. На рынке PLC доминируют проприетарные решения и технологии, а то, что продемонстрировали мы, являлось в большей степени демократичной открытой платформой, работающей на Ubuntu Core и обеспечивающей применение приложений PLC.

LXF: У Canonical весьма интересная история участия в MWC: на MWC 2015 они представили настольный коммутатор, запускающий самые разные сетевые приложения, и всё это было создано на дешевом оборудовании общего назначения. Затем на MWC 2016 они представили LimeSDR, который позволяет любителям и умельцам создавать собственные беспроводные устройства и протоколы. Не могли бы вы рассказать нам, что вы продемонстрировали на этот раз?

МБ: Оба этих проекта продолжают, кстати. В этом году мы показали базовую станцию LimeSDR,

работающую на платформе i7 от Intel. Это была базовая станция с диапазоном приблизительно в километр на плате PCI с чипсетом Lime. Мы также продемонстрировали использование Ubuntu Core с моментальными снимками — у нас была пара партнеров, которые поставляли LTE и стек Evolved Packet Core (EPC), и мы смогли показать, как они скачиваются на базовую станцию беспроводным способом, а затем запустить терминал LTE с базовой станции. Интересно, что посмотреть пришли практически все телекоммуникационные компании, присутствовавшие на MWC — их это очень заинтересовало. Сейчас EE делает пилотный проект с Lime. EE заявили, что их стратегия в Великобритании заключается в обеспечении 99% охвата, и чтобы этого добиться, они должны рассмотреть возможность инвестиции в совокупный расходный капитал. Итак, технологии типа Lime крайне интересны для EE и BT, и Vodafone тоже присматривается к ним. Это серьезные оценки технологии, и я думаю, что пока мы разговариваем, 5000 версий для разработчика LimeSDR как раз в процессе рассылки, так что это очень успешная краудфандинговая кампания. Разработчики смогут писать приложения для базовой станции, развертывать их с помощью наших инструментов Snapcraft, и это позволит создать опыт работы с базовыми станциями. По-моему, это действительно интересно, поскольку делает две вещи: во-первых, решает проблемы проприетарной природы и высокой стоимости, ассоциируемые с базовыми станциями, а во-вторых, обеспечивает место на рынке для поставки новейших приложений. Итак, при наличии открытой платформы мы можем поддержать оборудование Intel и ARM. Сейчас Lime производит архитектуру на базе Intel, однако не привязан к ней. У нас есть набор разработчика с USB-интерфейсом, чтобы вы могли заниматься разработкой на своем ноутбуке и подключить его как отдельную плату, а затем, когда вы развернете его, он развернется на PCI-карте. Так что это солидно уменьшит стоимость вывода базовой станции на рынок.

LXF: Можно предположить, что это вооружит также и независимых разработчиков, ведь им не надо будет покупать дорогостоящее оборудование или платить за доступ к проприетарным средам. Любой сможет прийти на праздник.

МБ: Верно. Это действительно интересно, и уже масса народу зарегистрировалось для получения набора разработчика, что показывает зарождающийся спрос на вхождение в область, которая прежде была проприетарной. И еще я думаю — благодаря тому, что мы не привязаны к любой конкретной технологии как таковой, будет возможно применять более привычные инструменты разработки с открытым кодом вместо сугубо проприетарного встроенного инструментария. Мы также демонстрировали пару сетевых коммутаторов на MWC, основанных на Facebook Wedge, это часть их Open Compute Project (OCP), так вот мы продемонстрировали моментальные снимки, работающие на устройстве OCP Wedge. И мы показали коммутатор Quanta, работающий на Ubuntu с моментальными снимками. Все они активны и дают

множество возможностей, о которых мне не рассказать, но среди них есть ряд действительно интересных для поставщиков сетевых услуг. Должен назвать эти организации несколько консервативными, но инициативы вроде OCP и Telecom Infra Project (TIP), появляющиеся на рынке, позволяют им начать мыслить, как сосредоточиться на программах, а не оборудовании. Технические гиганты, например, Facebook, хотят снизить стоимость оборудования, и наша инфраструктура снимков предоставляет для этого отличный способ, потому что мы можем надежно изолировать и поместить в контейнеры приложения, работающие на этом коммутаторе, на относительно эффективной операционной системе.

LXF: Я видел несколько интересных фото вашего стенда на MWC. Там был дружелюбного вида робот, который слонялся повсюду. Можете немного рассказать о нем?

МБ: Это сервисные роботы, один из них использовался в приложении в России, а другой — на Ближнем Востоке. Они работают на Ubuntu, их цель — обеспечивать услуги розницы, и привлечь их на стенд было забавно. Однако послание они несут вполне серьезное: что у Ubuntu есть множество приложений, готовых к развертыванию. Прямо сейчас в области робототехники Ubuntu по сути является платформой по умолчанию. Если вы посмотрите на комплект разработки Robot OS (www.ros.org), то он является вполне стандартным на Ubuntu, а для других ОС есть экспериментальные сборки. Итак, мы видим



» «Погружение через 5... 4... 3... 2...»

Один из примеров — шлем DAQRI, выставленный на вашем стенде.

МБ: ДР — очень популярная технология, но иногда можно сказать, что это технология в поисках проблем для решения. DAQRI — отличный пример находки хорошей ниши, предъявивший ремонтникам оборудования возможность вывести на дисплей добавочную информацию, дополнить их видение и помочь им решать сложные задачи. Есть ряд приложений — от аэрокосмических до предназначенных для ремонта и производства автотранспорта,

ОБ ЭТОМ РОБОТЕ

Прямо сейчас в области робототехники Ubuntu по сути является платформой по умолчанию.

огромный спрос от умельцев и очень серьезных крупных организаций по выводу на рынок роботов на платформе Ubuntu. Опять же, важно попасть на рынок вовремя: я могу использовать среды разработки и открытый код, чтобы поставить огромные количества, а потом они потенциально смогут добавить собственные проприетарные компоненты с закрытым кодом там, где базируются их IP, это их конкурентное преимущество.

LXF: Виртуальная реальность (и дополненная/смешанная реальность) обсуждается уже некоторое время, во многом ее рассматривают как игрушку, но, похоже, сейчас за нее взялись несколько серьезных проектов.

ремонта коммунальных служб, всего, где нужно правильно решить сложные проблемы. Он может обеспечить набор инструкций и совместить их. Там, где имущество чрезвычайно дорого — например, когда вы ремонтируете реактивный двигатель или автомобиль — устройства, подобные этому шлему, могут сыграть важную роль. Эти ДР-приложения, которые вы видите, продвигаются крайне медленно, потому что в такого рода средах всегда есть экономическое обоснование. Реальный конечный потребитель должен иметь предпосылки для покупки, что в основном обусловлено качеством. Но есть очень понятные случаи применения, способные создать отличные экономические обоснования. Если вы посмотрите, например, на какой-»

нибудь реактивный двигатель, которые сейчас в основном арендуются, время простоя для разработчика означает отсутствие дохода, так что это действительно интересное приложение.

LXF: По крайней мере, это означает, что у инженера свободны обе руки, а не заняты множеством щупов, соединителей и вилок.

МБ: Верно. Как показывает мой опыт, работа в таких областях, как нефть и газ, системы обеспечения и производство, всё это — области, где приходится иметь дело с исключительно дорогими активами, создаются они или ремонтируются. И возможность работать корректно и безопасно является основополагающей. Итак, есть много областей, где наличие подобного инструмента, высвобождающего руки, является критически важным.

LXF: Что из увиденного больше всего впечатлило вас на MWC?

МБ: Ничего особо сногшибательного, но кое-что всё же произвело впечатление. Одно — постоянное стремление к миниатюризации, и мы сами показывали множество подобных примеров на нашем стенде в плане размеров плат. Факт тот, что на платформах ARM и Intel, SoCs и платформы, которые мы можем поставить, становятся всё меньше и меньше, и стоимость снижается... Уменьшение и физического размера, и стоимости, и притом технологически возможности, умения устройства, которое стоит, скажем, менее \$10. Эта тема возникает повсеместно.

Поскольку это конференция, ориентированная на телекоммуникации, там было много 5G. Это меня немного смешит, поскольку нам бы надо уже полностью его стандартизировать. Однако энтузиазм провайдеров инфраструктур в демонстрации возможностей 5G действительно интересен. На самом деле он показывает, что есть альтернативные приложения, которые нельзя использовать при современных сотовых технологиях. В частности, речь идет не только о пропускной способности — речь идет о периоде ожидания. Одной

из целей 5G является период ожидания в субмиллисекунды, и я думаю, что это делает область приложений очень интересной, в частности с точки зрения телекоммуникаций.

Большое и больше Интернета Вещей, и, по-моему, большое внимание также уделяется технологии Low Power Wide Area Network (LPWAN). Не думаю, что эти технологии еще будут живы через пять лет. Сейчас у нас есть NarrowBand IoT (NB-IoT), который поддерживается GSM, а также LTE-M, у нас есть SIGFOX и LoRa. Они очень интересны с точки зрения IoT, но я не уверен, что телекоммуникации заработают на этом столько денег, сколько ожидают, потому что это технологии с относительно низкой скоростью передачи данных. Но для производителей устройств они очень интересны, потому что мы можем, например, получить соединение с обогревательным котлом, помещенном в подвале, откуда обычное сотовое соединение было невозможно. Такие технологии с низким энергопотреблением, определенно, лучше проникают в труднодоступные и удаленные места, и они весьма интересны с этой точки зрения. В последнее время много говорят о беспилотных автомобилях, и инвестиции в область исследования и разработки в этой области огромны. Мы видим, как автомобильные компании движутся к миру ПО, и это приведет к появлению самых сложных устройств.

LXF: Традиционные провайдеры услуг телекома в наши дни сталкиваются со множеством проблем. Недавно Canonical объявил о партнерстве с AT & T, и это важный шаг. Каким образом это и другие партнерства Canonical помогут модернизировать «старую гвардию»?

МБ: Подразделение Облачных технологий в Canonical добилось больших успехов в области телекоммуникаций. Они поставляют такие технологии, как OpenStack, Metal as a Service (MaaS), Kubernetes, они отлично соотносятся с телекоммуникациями. Им нужно освежить свою инфраструктуру движка, максимально ее автоматизировать. Раньше большим достижением была виртуализация, и одной из самых больших проблем Network Function Virtualisation (NFV) была работа с ней. Поэтому в действительности то, что мы описываем, было целым пакетом технологий, и именно это используют



► Угадываете, что оно делает? Ответы — на открытке в Башни Linux Format...

такие компании, как AT&T, снижая стоимость работы. Это действительно большой скачок в облачной области для телекоммуникаций. Подразделение устройств и IoT, которое я возглавляю, этот фокус весьма интересен, потому что он во многом передовой — мы говорили о линейной базовой станции — но также попадает прямо в дом. Сейчас некоторые из телекоммуникационных компаний имеют проблемы с роутерами DSL, которые имеют «белую» метку и никогда должным образом не рассматривали проблемы безопасности. Мы видим возможность для Ubuntu Core с точки зрения безопасности дома и на малых предприятиях, но также возможность работать на платформе. Прямо сейчас ваш обычный Wi-Fi-роутер делает не очень много, и одной из вещей, которую мы исследуем вместе с телекоммуникациями — предложение добавочных сервисов на этом оборудовании. Итак, вместо того, чтобы считать его сугубо специфическим устройством, как насчет того, чтобы вложить туда немного больше памяти и вычислительной мощности, и тогда мы сможем запускать моментальные снимки, которые позволят телекоммуникационным компаниям предлагать дополнительные платные услуги. Или бесплатные услуги, помогая снизить отток абонентов и розничных клиентов.

LXF: И в заключение, не могли бы вы подвести итог общей стратегии и целей Canonical для этого пост-настольного мира?

МБ: Всё внимание Canonical сфокусировано на Software Defined Everything [ПО определяет всё]. Это простая фраза, но по сути она означает нечто очень мощное: тот факт, что ПО меняет мир. Делая его с открытым кодом и давая ему возможность работать на любом устройстве — будь то небольшое устройство дешевле \$10 или работающий на GPU сервер в центре обработки данных — это ПО имеет решающее значение. Это ПО меняет предприятия, и именно это мы хотим обеспечить. **LXF**



► Если у устройства меньше четырех антенн, то оно не суперсовременное.

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Смерть патента

Чудеса всегда случаются.
Онидзука. Эпизод 28

Патенты — это отрывка цивилизации. Они возникли на довольно ранней ее стадии, когда в Венецианской республике XV в. пытались нивелировать параноидальное сокрытие технологических тайн со стороны цехов и гильдий. Прошло уже почти 600 лет, цеха и гильдии давно уже не заправляют миром, но человечество до сих пор «рыгает» патентами. Кстати, на момент своего возникновения время действия патента было ограничено десятью годами! Жизнь с тех пор существенно ускорилась, но не в области патентного законодательства.

Формат сжатия звука MP3 разработали в институте Фраунгофера. Алгоритм был опубликован в 1993 г. и довольно скоро стал популярным: иски тогда были не резиновые, а музыке с компьютеров слушать хотели буквально все. В 1998 г. институт разослал «письма счастья» на тему, что за использование патентованной технологии надо платить. В принципе это естественно — люди просто хотели получить денег с внезапно популярного алгоритма; но этому желанию потворствовали аж 24 года!

История эта закончилась: срок патента истек в апреле. Теперь в основных дистрибутивах GNU/Linux MP3 будет «из коробки». Не особое оно достижение, но одним неудобством меньше. Это не первый патент, который экосистема GNU/Linux пережила, и далеко не последний.

PS Вышел INSTEAD 3 — читателям и писателям посвящается.
E.m.Baldin@inp.nsk.su



В этом месяце вы научитесь...

- Скрывать информацию** 54 **Виртуализировать** 58
Нейт Дрейк обзавелся отличной программой *Outguess*, и уж теперь-то никто не доберется до портретов его котятры. Вам, конечно, известны *VirtualBox* или *Vmware*, но **Стюарт Бернс** познакомит вас еще и с платформой Proxmox с *KVM* поверх нее.
- Плясать с Samba** 62 **Ругать ключами** 66
 Присоединяйтесь к **Джонни Бидвеллу** — получится зажигательный сервер. Особенно если управлять настройкой через *Webmin*. В заботе о новичках, **Джон Лэйн** подробно описывает создание пары ключей шифрования и учит управлять этими ключами.
- Ускорять сайты** 70 **Воспитывать свой дом** 74 **Создавать игры** 78
 Сайт у вас тормозит? А у **Михалиса Цукалоса** — нет! Зачем тратить время на обработку повторных запросов, лучше запасти ответ в кэше. **Максим Черепанов** продолжает добиваться от своего дома беспрекословного послушания, организуя взаимодействие файлов и процессов. **Михаилу Вознесенскому** попался на глаза замечательный игровой движок *socos2d*; писать игры с таким — сплошное удовольствие.

АКАДЕМИЯ КОДИНГА



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

- И сразу в Google Play** 82 **Лучше увидеть** 92
 При помощи Android Studio **Кент Ельчук** разрабатывает приложения, которые годятся в дело с ходу. Однообразные с виду множества данных **Михалис Цукалос** превращает в осмысленные графики.
- Ищите и обрящите** 88 **Мандельброт и другие** 96
 Утилита *find* существует и без нас, но **Михалису Цукалосу** захотелось написать свою. Как математик, **Михалис Цукалос** приходит в восторг от стройной красоты фракталов.

Stego: Outguess

Скройте свои личные данные в обычных файлах с помощью этой прекрасной стеганографической программы и человека с интересным именем **Нейт Дрейк**.



Наш эксперт

Нейт Дрейк — независимый журналист, специализирующийся на кибербезопасности и ретро-технологиях.

Еще в 440 году до н.э. тиран Гистей, зная, что его сообщения перехватывались, обрил голову своему самому доверенному рабу (это, понятное дело, оксюморон [Ред.: — И очень глупый]) и вытатуировал на этой голове сообщение своему вассалу Аристагору. Времени тогда явно хватало, поскольку рабу разрешили дожидаться, пока его волосы отрастут, после чего отправили его с безобидным на вид письмом в руках. По прибытии раб попросил Аристагора обрить его и прочесть реальное послание, поручавшее вассалу восстать против персов.

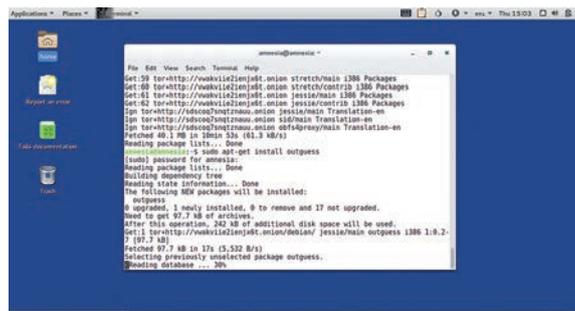
Если на миг отвлечься от того факта, что было бы куда проще велеть рабу сообщить секретное послание устно [Ред.: — Не слишком-то и доверял?], это первое зафиксированное использование стеганографии, когда секретное сообщение скрыто внутри другого, с виду обычного.

В современном мире, опутанном проводами, где у нас нет времени сидеть и смотреть, как растет щетина, стеганография (часто сокращаемая до 'stego') осуществляется с помощью различных компьютерных программ. Программ для стего немало, но все они работают по примерно одному принципу. Сначала создается секретный файл, например, текстовый документ с сообщением. Затем он помещается в безобидный с виду файл-контейнер, например, в изображение вашего котеночка.

Некоторые файлы, например, изображения, содержат случайный, казалось бы, цифровой «шум», вызванный внешними факторами, такими как датчик камеры или какой-либо способ сжатия. Инструменты стего используют этот шум для сокрытия изображений внутри него. Очень сложно проанализировать данные, чтобы определить разницу между файлом с многообразными видами «шума» и файлом стего, содержащим скрытые данные.

Коварный стегоанализ

Многообещающе выглядящий на бумаге стего в прошлом был чем-то вроде гонки цифровых вооружений, с инструментами, так же быстро разрабатываемыми для сокрытия изображений, как



Для установки *Outguess* используется Tails OS. Посетите сайт <http://tails.boum.org>, чтобы установить его на USB или DVD.

и программы, созданные для обнаружения использования стеганографии в файлах.

В 2001 г. был всплеск беспокойства по этому поводу, когда на новостных сайтах типа USA Today сообщалось, что террористы обменивались изображениями, размещенными онлайн в спортивных чатах, сайтах для взрослых и досках объявлений.

Исследователи потратили месяцы напряженных исследований на эту тему. Короче говоря, очень трудно доказать обратное, т.е. что любой файл категорически не содержит тайное послание.

Конечно, это работает в обе стороны. С правильными инструментами и крупицей здравого смысла вы можете использовать стеганографию, чтобы защитить свои самые темные секреты, обмениваясь внешне безобидным медиа-файлами через Интернет.

Вы можете удивиться, почему мы выбрали для защиты данных именно это, вместо использования обычных программ шифрования. А дело в том, что каждый раз, когда вы отправляете прикрытый паролем файл или почту своему контакту, содержание отправления, может быть, и нельзя угадать, но любой перехвативший ваше сообщение смекнет: вы что-то скрываете, и придется разбираться дальше.

Во время холодной войны шпионы оставляли и забирали пакеты в определенных «тайниках», согласованных с контактом. Подходило любое место, и не требовалось, чтобы шпион и связник одновременно посещали тайник.

Стеганография может работать как цифровой тайник. Если, например, вы скрываете свои файлы внутри изображения и затем загружаете это изображение на сайт, вам даже не надо быть в Сети одновременно с вашим контактом, да и вообще незачем знать друг друга.

TAILS и Outguess

Для этого учебника была выбрана *Outguess* — отличная утилита командной строки Нильса Провоса [Niels Provos], инструмент стего. Прежде всего потому, что она доступна для установки из репозитория Ubuntu/Debian; ну и очень проста в использовании.

Хотя этот инструмент какое-то время не обновлялся, будущее у него есть: он позволяет скрыть два сообщения внутри файла-контейнера с отдельными паролями, работая аналогично функции

Скорая помощь

Утилита командной строки *stegdetect* используется для обнаружения стего в файлах. Однако в наших тестах она не смогла обнаружить наш стего, закодированный с использованием *Outguess*.



Статья в USA Today, утверждающая, что бен Ладен использовал стего. Нильс Провос, создатель *Outguess*, изучил тысячи изображений в Интернете и не нашел тому доказательств.

Уклонись и раскройся

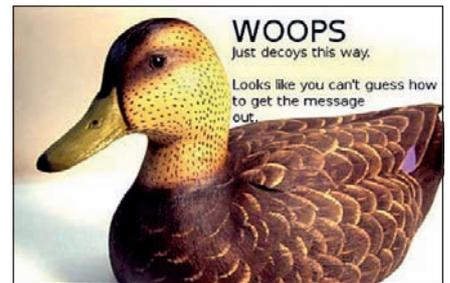
Лет пять назад таинственная организация под названием Cicada 3301 начала публиковать в Интернете серию всё более и более затруднительных головоломок, чтобы нанять «очень умных людей». Первой головоломкой 2012 г. было изображение с небольшим текстовым файлом в конце. Текст был закодирован простым шифром Цезаря, когда каждая буква алфавита смещается на определенное количество букв (в данном случае — на четыре).

Это приводило к изображению деревянной утки с надписью "Woops! Just decoys this way. Looks like you can't guess how to get the message out [Ync! Вот такая маскировка. Похоже, вы не угадаете, как выудить сообщение]". Сметливые крипто-ботаники всего мира объединились в коллективного гения и поняли, что означают слова "guess" и "out": в изображении скрыто сообщение с помощью *Outguess*.

Решение этой головоломки привело к появлению другой, на странице Reddit, основанной на цифрах мая. Поскольку для перехода к следующей головоломке надо решить предыдущую, лишь горстка людей дерзает утверждать, что они решили задачи Cicada 3301; но их заявления трудно проверить, поскольку решения были опубликованы.

Мы практически ничего не знаем о том, кто такие Cicada 3301 или почему они пытаются нанять одаренных людей. Задачи в последующие годы становились всё сложнее, включая QR-коды на фанерных столбах, расположенных на протяжении от Польши до Гавайев, а также заказные сайты и записанные телефонные сообщения.

Около года назад головоломки прекратились, но те, кто захочет увидеть старые задачи, могут посетить <https://cicada3301.org>.



➤ Исходное изображение утки содержало два сообщения, закодированные с *Outguess*. Одно из них была цифровой подписью, а другое вело к /r/a2e7j6ic78h0, таинственному сообществу Reddit, с новой загадкой.

правдоподобного отрицания в таких программах шифрования, как *VeraCrypt*. Если когда-либо найдется способ обнаружить стего в ваших файлах, вы можете назвать ложный пароль, чтобы открыть заранее подготовленную поддельную информацию.

Одним из способов обнаружения стего является анализ жесткого диска на наличие *Outguess* или проверка интернет-истории, чтобы узнать, загрузили ли вы его. По этой причине мы рекомендуем для данного проекта использовать операционную систему TAILS.

TAILS загружается целиком в ОЗУ, поэтому все следы активности теряются через несколько минут после выключения компьютера, в том числе файлы, которые вы использовали, и программа *Outguess*.

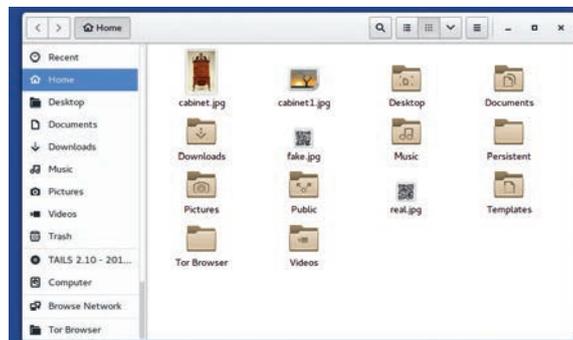
Другим преимуществом TAILS является то, что он маршрутизирует все соединения через сеть Tor. Это делает практически невозможным кому-то отслеживать ваши соединения, чтобы сообщить, что вы скачали TAILS, или куда вы загрузили свой файл-контейнер со скрытым сообщением.

Вступаем на путь фальшивок

В целях этого проекта представьте, что вы урвали заветную *Bill and Ted's most Excellent Collection* на DVD, которая поставляется в комплекте с обоими фильмами и дополнительным диском с видеоинструкцией о том, как профессионально играть на воздушной гитаре.



➤ Здесь стего-изображение доступно под видом предмета для продажи на интернет-сайте.



➤ Здесь два файла *real.jpg* и *fake.jpg* были встроены в изображение *cabinet.jpg*, чтобы создать новый файл *cabinet1.jpg*.

Скорая помощь

Вы можете загрузить DVD-образ TAILS с <http://tails.boum.org>. В противном случае вставьте DVD с Ubuntu и выберите «Попробовать без установки» в меню загрузки.

Скорая помощь

Чем больше ваши секретные файлы, тем больше «шума» вам понадобится. Для получения наилучших результатов старайтесь использовать большие изображения или аудиофайлы.

Вы справедливо обеспокоены тем, что эту ценную собственность могут украсть, и решаете закопать комплект DVD, чтобы его можно было вырыть и прибыльно продать лет через 20.

Спрятав комплект, вы решаете сообщить место доверенному родственнику — на случай, если сами не сможете его забрать. Вы приватно встречаетесь с ним, чтобы договориться об используемом пароле и о том, что вы скроете описание этого местоположения внутри изображения антикварного шкафа в стиле чиппендейл.

Поскольку вы оба живете в полицейском государстве [Ред.: — Океании?], где правительство жаждет завладеть классикой 1990-х годов, вы заодно договариваетесь о подставном пароле, который под нажимом любой из вас может ввести, чтобы дать ложное местоположение.

Вы оба также договариваетесь применять Live DVD для установки *Outguess* и кодирования/декодирования файлов, чтобы обеспечить на ваших машинах отсутствие следов инструмента.

Вы решите использовать QR-код для хранения GPS-координат как реального местоположения набора DVD, так и поддельного, поскольку их легко сканировать в телефон и открыть в таких приложениях, как Google Maps.

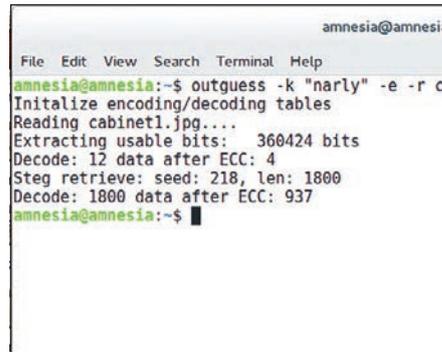
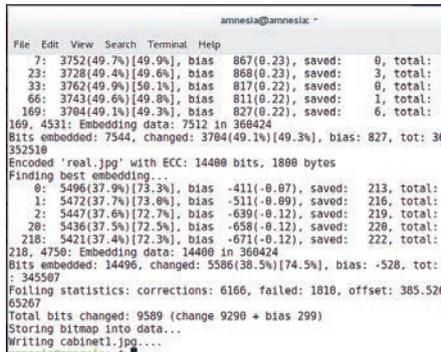
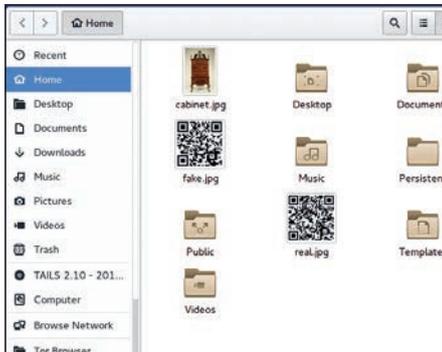
Когда стего-файл будет подготовлен, ваш друг сможет получить его в любой момент, чтобы выяснить, где спрятан фильм Билла и Теда.

Стего дампа

Хотя использование *Outguess* в live-версии Linux, скажем, такой как TAILS, приведет к удалению всех следов инструментов и файлов,

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

Стеганография в три простых шага



1 Установите Outguess

Загрузите выбранную вами ОС (в нашем случае идеальна TAILS) и откройте терминал. Запустите `sudo apt-get update`, а затем выполните `sudo apt-get install outguess`. Пока идут обновление и установка, займитесь перемещением файла внешнего контейнера, а также реального и поддельного файлов в домашнюю папку. Не бойтесь называть эти файлы вполне явно, типа "real" и "fake", поскольку *Outguess* не хранит исходные имена файлов.

2 Закодируйте файлы

Предположим, контейнерный файл называется `cabinet.jpg`, а реальный и поддельный файлы называются `real.jpg` и `fake.jpg` и закодированы с паролями "narly" и "bogus" соответственно. Выходной файл называется `cabinet1.jpg`. В этом случае команда для сокрытия файлов, будет выглядеть так: `outguess -k "bogus" -d fake.png -E -K "narly" -D real.png cabinet.jpg cabinet1.jpg`. При создании собственных стего можно свободно изменить любое из этих значений.

3 Раскодируйте файлы

Как только ваши файлы будут загружены, ваш контакт должен снова повторить шаг 1, затем загрузить стего-файл. Если он действует под контролем или нажимом нежелательных лиц, то может запустить эту команду с ложным паролем, чтобы открыть фальшивый файл `decoded.jpg`: `outguess -k "bogus" -r cabinet1.jpg decoded.jpg`. В противном случае можно открыть реальный файл таким образом: `outguess -k "narly" -e -r cabinet1.jpg decoded.jpg`.

с которыми вы работали, исходные файлы секретов и контейнеров не удаляются. В идеале надо создавать их тоже в live-системе; в противном случае, закончив, примените команду `shred` ко всем соответствующим данным.

Важный момент: используете ли вы изображение, звуковой файл или видео, обязательно запишите его сами. Если копия исходного «необработанного» файла доступна онлайн, ее можно сравнить с вашим, и применение стеганографии будет обнаружено. Решив загрузить серию фотографий одного и того же предмета, например, антикварного шкафа, помните, что все они должны быть примерно одного размера.

При согласовании пароля с вашими контактами разработайте систему, по которой они смогут узнать, где найти файлы-контейнеры. Например, можно каждую неделю публиковать в оговоренное

время определенное объявление, или указать определенный номер телефона рядом с каждым изображением, содержащим скрытые файлы.

Многие сайты сжимают или иным образом изменяют медиа-файлы после загрузки. Попробуйте найти такой, который позволяет загружать файлы без изменений, или разместите ссылку на загрузку оригинала.

Если вы живете под юрисдикцией страны с законами об обязательном раскрытии информации, такой как Великобритания, то технически вы должны по требованию предъявить все ключи, находящиеся в вашем распоряжении. Это включает как поддельный пароль, так и реальный. Изыщите время, чтобы проверить законность использования стего и сокрытия реального пароля там, где вы живете. **LXF**

Аутентификация сообщения

В том, что вам не надо встречаться с вашими контактами, чтобы отправлять секретные стего-сообщения, есть и сильная, и слабая сторона. Если противник обнаружит ваш стего-файл, он может заменить его одним из своих, с ложной информацией, пытаясь заманить в ловушку вас или ваших контактов.

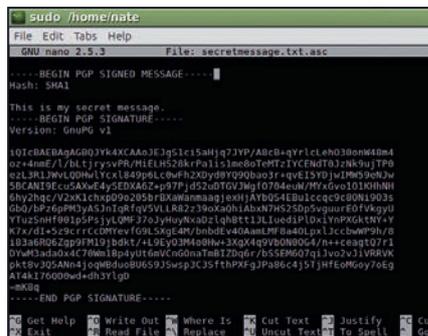
Поэтому рассмотрим применение утилиты командной строки *gpg* для цифровой подписи всех отправляемых сообщений. Эта утилита традиционно используется для шифрования данных — файлы закодированы с помощью открытого, или публичного, ключа, который вы предоставляете всем, и декодируются отдельным закрытым, или приватным, ключом, который никогда не покидает ваш компьютер.

Однако этот процесс может работать в обратном порядке, когда вы кодируете файл своим закрытым

ключом. Это не гарантирует безопасность передачи данных, поскольку файл может прочитать любой, у кого есть ваш открытый ключ; зато позволяет вашему контакту убедиться, что сообщение отправили именно вы.

Подробную инструкцию по использованию *gpg*, предустановленного в Linux, можно найти на сайте www.gnupg.org/documentation/manpage.html. Любая попытка изменить содержимое файла приведет к «поддельной» подписи.

Всякий раз, когда вы велите *gpg* подписать файл, создается новый файл с расширением `.asc`, который затем можно спрятать внутри файла контейнера с помощью *Outguess*. Это работает для всех типов файлов. Когда ваш контакт получит подписанный файл `.asc`, попросите его запустить команду `gpg --output filename.jpg --decrypt filename.jpg.asc`, где `filename.jpg.asc` — извлеченный файл.



Текстовый файл, подписанный с *gpg*. Если противник изменит хотя бы одну букву вашего сообщения, подпись проверку не пройдет.

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!



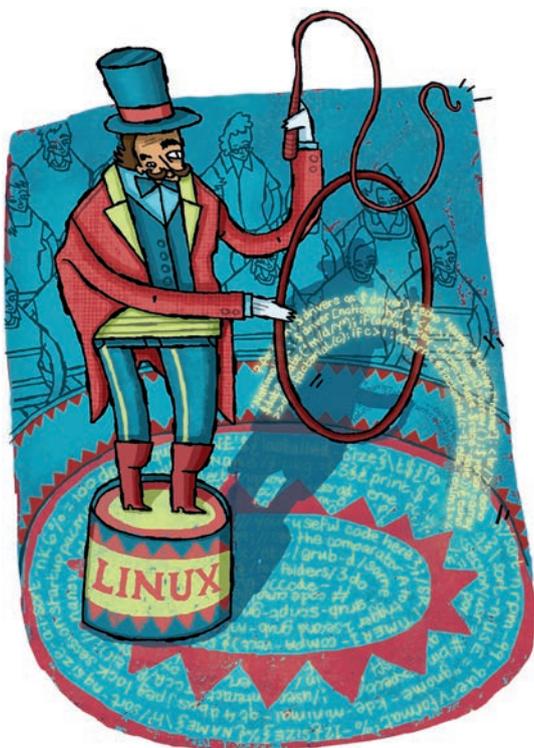
ProxMox: Гипервизор, тип 1

Стюарт Бернс помогает вам сделать первые шаги в области серверов виртуализации, пользуясь платформой виртуализации ProxMox.



Наш эксперт

Днем **Стюарт Бернс** является экспертом крупномасштабной инфраструктуры виртуализации. А ночью подрабатывает администратором Linux: ему нравится разбираться с проектами.



Многие читатели знакомы с такими продуктами виртуализации рабочего стола, как *VirtualBox* или *VMware Workstation*. Эти продукты виртуализации называются «гипервизорами второго типа» и обычно функционируют

на операционной системе бок о бок с обычными приложениями. Они удобны в качестве решений, но не лишены ограничений. Они не предназначены для запуска приложений 24/7. Не в последнюю очередь потому, что если придется перезагрузить ПК, на котором они работают (или еще хуже — если ПК сломается), все виртуальные машины тоже надо будет перезагружать.

А вот гипервизоры первого типа предназначены для работы на специализированном оборудовании, которое по возможности должно приближаться к чисто аппаратной платформе, для обеспечения наилучшей производительности и стабильности. ProxMox является гипервизором первого типа (так же, как VMware ESXi и Hyper-V).

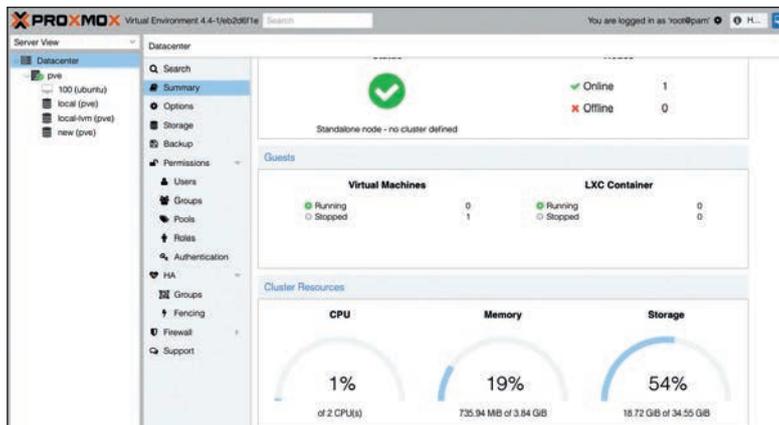
ProxMox — платформа гипервизора на базе Debian, с пакетом виртуализации KVM поверх и простым в использовании GUI, дополнительно облегчающим жизнь. ProxMox также поддерживает контейнеры, если они необходимы. ProxMox бросается в глаза как идеальный выбор новичка в Linux, потому что большинство задач администрирования не требуют обращения к командной строке. Хотя под капотом также доступен набор инструментов командной строки KVM через SSH. Все функции доступны в свободной версии, однако мы рекомендуем оказать поддержку всем прелестям свободного ПО, по возможности приобретая контракт на поддержку.

Для данного введения в ProxMox читателю понадобится относительно современный ПК или сервер как минимум с 1 ГБ ОЗУ. В реальном мире, для чего-то полезного, более разумной стартовой точкой будут 4 ГБ. ProxMox также требует два жестких диска разумного размера. Понадобится еще прожечь носитель с записанным ProxMox 4.4 Installation. Его можно скачать с www.proxmox.com/en/downloads вместе с различной документацией.

Установка гипервизора

Чтобы начать установку сервера ProxMox, загрузитесь с установочного CD-ROM 4.4, записанного ранее, и следуйте шагам по установке. Начальный экран установки представляет ряд опций. Для целей нашего урока выберите уже выделенный пункт, Install ProxMox VE, или просто нажмите Enter. На открывшемся экране примите лицензию, нажав на кнопку 'I agree' справа.

Следующий экран относится к настройке жесткого диска. ProxMox установится на первый же доступный диск (обычно это диск sda). Есть расширенные опции, подлежащие настройке, такие как формат или структура диска. Настройка по умолчанию — Ext4, но доступны и другие опции, включая ZFS и переразбивку диска. В рабочей среде устройство загрузки должно быть зеркально отобразено, дабы предотвратить сбой, который приведет к остановке работы хоста и, следовательно, виртуальной машины. Всё готово? Жмите на следующую страницу мастера.



➤ Вид DataCenter. ProxMox не только бесплатен, он еще и красив, и имеет продвинутые функции, которые иначе стоили бы немалых денег и отчислений за лицензию.

Контейнеры тоже полезны!

Помимо привычных виртуальных машин, подобной той, что мы создали в этом уроке, ProxMox также поддерживает контейнеры. Linux Containers (LXCs) позволяют работать с несколькими серверами или сервисами изолированно, но использовать один и тот же виртуальный хост, экономя таким образом большое количество памяти и прочих системных операций, которые вовлекла бы классическая виртуализация.

Linux Containers обеспечивают полуизолированную среду, где можно запускать серверы. Контейнеры подходят не для всего, но могут быть идеальны для запуска сервисов. Контейнеры также полезны, потому что они исключительно портативны, и перемещать их между серверами очень легко. Linux Containers также устраняют множество проблем с настройкой, имеющих место при развертывании классических VM.

Эти сервисы работают на сервере. Подробности вы найдете на официальном сайте проекта Linux Containers <https://linuxcontainers.org/>. Чтобы попробовать эту опцию, используйте кнопки Template [Шаблон] и Create container [Создать контейнер] рядом с задействованными опциями VM. Опций настройки множество, но я рекомендую просто поэкспериментировать и посмотреть, что есть что и как работает.

Установка ProxMox предполагается простой, беспроblemной и быстрой. Вопросов задается очень мало, и все они очевидны. Одна странность заключается в том, что при поиске клавиатуры UK вы найдете ее под названием Britain, а не United Kingdom. Программа установки задаст вам несколько вопросов, в том числе о стране, адресе электронной почты и настройке сети. После указания страны автоматически определятся часовой пояс и раскладка клавиатуры. Адрес электронной почты должен быть реальным, иначе вы не сможете получить учетную запись поддержки ProxMox.

На последнем экране, который требует вашего вмешательства, администратору надо настроить IP-адрес, DNS и имя хоста. По умолчанию опции сети будут по возможности заполнены полученной от DHCP настройкой. Настоятельно рекомендуем использовать статический IP-адрес. Также рекомендуем заранее настроить DNS для сервера — настройку по умолчанию следует изменить. Введите свои настройки сети поверх полученных от DHCP. И, наконец, когда установка завершится, последний экран предложит пользователю перезагрузиться.

Администратору также надо будет задать пароль для root. По завершении установки системы ее надо перезагрузить. После перезагрузки администратор может перейти на IP-адрес, указанный на экране консоли.

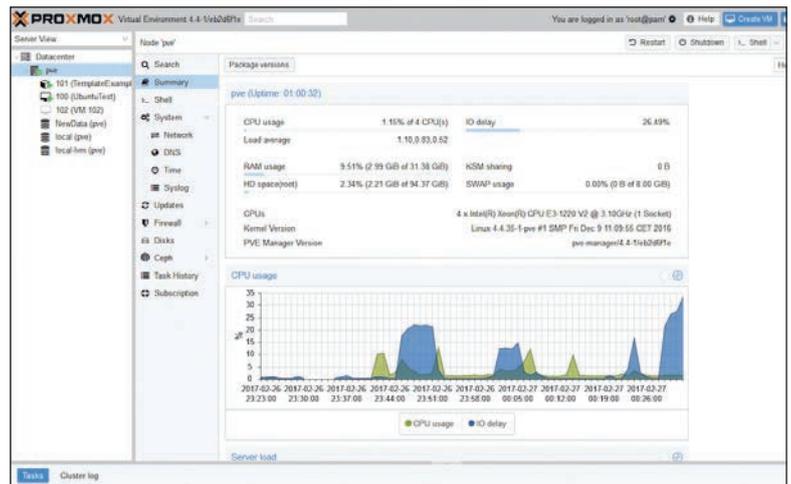
Зайдите на этот URL с помощью HTTP через порт 8006. Например, наш URL — <https://10.0.0.40:8006>.

На экране приглашения введите имя пользователя 'root' с паролем, который был задан во время установки. В зависимости от конфигурации вашей системы, вы можете получить предупреждение сертификата безопасности браузера. В данном примере нормальным будет принять самоподписанный SSL-сертификат.

При желании сервер ProxMox может использовать аутентификацию LDAP/AD, но она вне рамок данного введения в ProxMox. Итак, добро пожаловать в консоль управления, где выполняется большая часть задач администрирования!

В этот момент появится уведомление для пользователя о том, что отсутствует контракт на поддержку. Не волнуйтесь по этому поводу: ProxMox можно свободно использовать так, как вам хочется. Никакой контракт не обязателен. Тем не менее, контракт на поддержку может просто быть полезным, и ваше желание заплатить всегда приветствуется.

Чтобы понять, как работает консоль, знайте, что имеется несколько разных видов отображения. Для изменения текущего вида выберите требуемый вид из выпадающего списка видов в верхнем левом углу. В зависимости от вида, в крайней левой колонке будут появляться хосты, диски, кластеры. Справа от нее находится область меню, состоящая из ленты вверху и набора гиперссылок слева. Именно в этой части окна и делается большая часть настройки.



➤ Очень легко сразу же увидеть, что происходит с диском и памятью на вашем физическом сервере.

Важно отметить, что выпадающее меню в левой части отображает разные опции и настройки в зависимости от выбранной опции. Следовательно, если некий пункт отсутствует, дважды проверьте используемый вами контекст/вид.

Виртуальные новички

ProxMox отличается большой гибкостью и может использоваться как в отдельной, независимой [standalone] системе, так и в кластерах, которые используют хранилище общего доступа и несколько хостов ProxMox с такими функциями, как живая миграция и HA (High Availability — высокая доступность), где в случае сбоя одного из хостов задействованные VM перезапускаются на другом доступном хосте.

По умолчанию один хост, который мы установили ранее, был настроен, и появится среди объектов root под названием 'DataCenter'. Дополнительные хосты (если таковые добавляются) размещаются под объектом DataCenter.

Обратите внимание, что несколько хостов ProxMox можно объединить в кластер и использовать общее хранилище, для обеспечения высокой доступности. Кластер требует дополнительной настройки.

Лучше всего после установки проверить, обновлена ли операционная система хоста. Это можно сделать, перейдя на хост ProxMox в виде сервера (не в виде DataCenter) и выбрав Upgrade в левом меню. Обновления повысят уровень хоста ProxMox до самой последней редакции сообщества. Заодно установятся приложения вроде *parted*, которые в дальнейшем понадобятся на нашем уроке, так что, пожалуйста, не пропускайте этот шаг.

Скорая помощь

Одновременно с системой установите SSH Server. Удаленное управление вашими виртуальными серверами облегчит вам жизнь. Возможно, администратору придется провести дополнительную настройку.

»

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

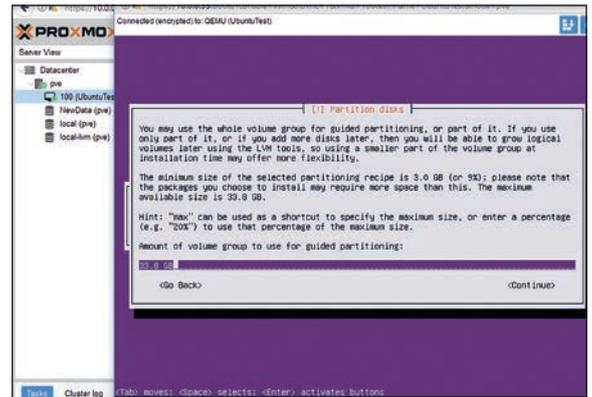
Добавление пользователей ProxMox

Использование учетной записи root — не самая мудрая идея, по разным причинам. Для каждого пользователя-администратора рекомендуется создать нового пользователя. В нашей системе база данных пользователей хранится локально. Добавьте нового пользователя-администратора, выбрав Permissions [Разрешения] в левом меню, Permissions > Users [Разрешения > Пользователи] и затем Add [Добавить] из строки меню. Добавьте Unix-имя нужного пользователя, настоящее имя и прочую необходимую информацию. Оставьте групповые и настройки области, как есть. Нажмите на Add, чтобы добавить пользователя.

Роли распределяются с помощью разрешений. Выбрав пункты root DataCenter в левом меню, выберите Permissions > Add. Чтобы предоставить права администратора новому пользователю, выберите из ленты Add > User Permissions [Добавить > Разрешения пользователя]. Путь должен быть установлен на '/', т.е. корневой каталог. Выберите пользователя из выпадающего окна, а затем роль. Роль администратора предоставляет полные права администратора. При необходимости можно создать индивидуальные роли. Администратор также может задавать права на уровне папки.

Настройка локального хранилища

По умолчанию загрузочный диск ограничивается первым диском, при этом под виртуальные машины выделено очень мало места, независимо от реального объема диска. Администратору понадобится использовать второй диск. Однако на первом диске есть место для установки очень маленькой (до 10 ГБ) VM, если надо. В показанной конфигурации мы добавили второй физический диск,



➤ При установке выбранного вами дистрибутива (здесь — Ubuntu) вам придется разбивать свой виртуальный диск на разделы.

чтобы хранить на нем виртуальные машины. ProxMox поддерживает ряд подсистем хранения, в том числе iSCSI, NFS и прочие инфраструктуры. Нельзя добавлять неподготовленные или отформатированные локальные диски. Все локально подсоединенные диски должны быть настроены на использование LVM (Logical Volume Management — управление логическими томами).

Настройка второго диска потребует немного поработать в командной строке. Чтобы добавить второй диск к хосту ProxMox, зайдите через ssh на сервер ProxMox от имени root (логин root по умолчанию разрешен), используя учетную запись и пароль root. В этой ситуации администратор может использовать инструмент *GParted* для создания второго диска с LVM. Чтобы создать этот второй диск, используйте приведенные ниже команды. Предполагается, что второй диск — **sdb**, но у вас это может быть по-другому:

```
ssh root@10.0.0.40 (Подстановка IP-адреса вашего ProxMox и пароля, заданного при установке)
parted --script /dev/sdb mklabel gpt
parted -- /dev/sdb mkpart primary ext2 1 -1
parted /dev/sdb set 1 lvm on
vgcreate newData /dev/sdb1
```

Если вы получили сообщение об ошибке, что *parted* не установлен, можете установить его с помощью команды

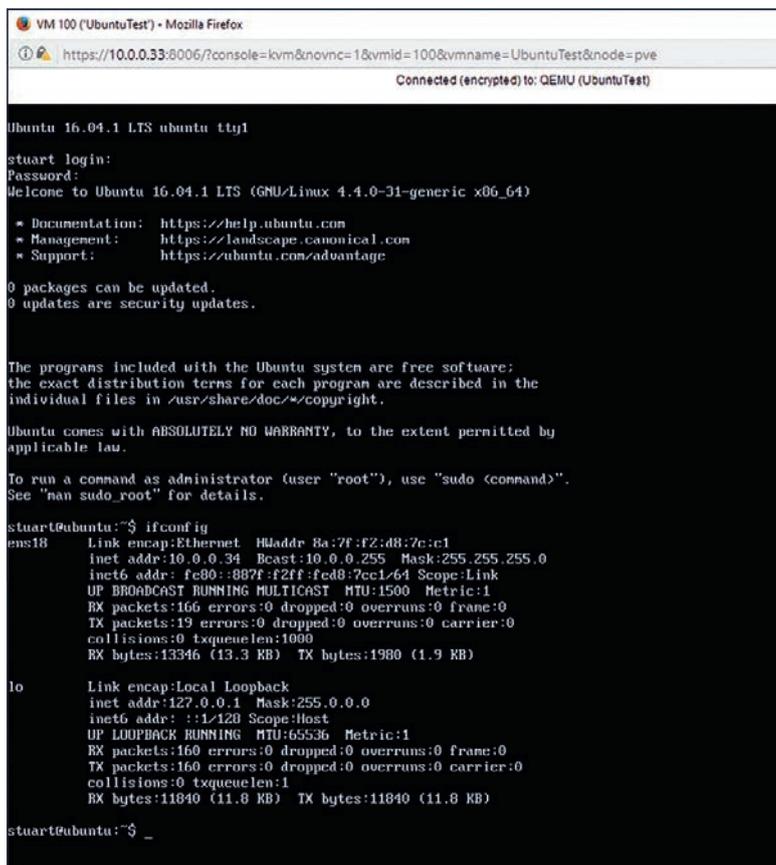
```
apt-get -y install parted
```

Разбив диск на разделы, воспользуйтесь командой *pvcreate*, как показано ниже, для создания физического тома (PV) на диске LVM:

```
pvcreate /dev/sdb1
```

В этот момент свеженастроенный LVM-диск можно сделать доступным для ProxMox System для хранения. Чтобы добавить новое хранилище, выберите вид DataCenter и затем на ленте кнопок выберите Add. Она раскроется, представив ряд возможных опций. Выберите LVM из выпадающего списка. Первое поле — для человеко-читаемых имен (лучше всего делать их короткими и внятными. Пробелы не допускаются.). Наконец, выберите вновь созданный диск в выпадающем поле группы тома. Вновь созданный LVM-диск должен появиться в выпадающем списке. Как только вы нажмете OK, хранилище добавится на сервер. Можно также создать область хранения специально для ISO-файлов.

По умолчанию вновь созданный диск может хранить виртуальные машины и контейнеры, но можно использовать любой доступный том. Чтобы ограничить использование хранилища до определенных типов, откройте выпадающий список контента и выберите нужные опции. В данном примере мы оставляем их по умолчанию. В зависимости от формата хранения, имеются дополнительные опции, например, ISO, резервные копии, и т.д.



➤ Запуск Ubuntu в среде виртуализации ProxMox. Производительность достаточно неплоха, но везде, где можно, надо использовать SSH.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Облегчите себе жизнь с помощью шаблонов

Развертывание нескольких виртуальных машин по методу, предложенному в нашем введении, займет много времени — времени, которое лучше потратить на другое. К счастью, массу времени экономят шаблоны. Шаблон машины означает, что вы можете создавать новые машины быстро и с минимумом ручного вмешательства. К счастью,

создание шаблонов — процесс несложный. Щелкните правой кнопкой по VM, чтобы создать шаблон — и вы можете конвертировать эту машину в клон (она станет недоступной в качестве работающей VM) либо создать клон как новую машину.

При использовании шаблонов VM администратору рекомендуется установить и настроить только

те сервисы и обновления, которые необходимы для создания солидной базовой установки VM; а уже в ней можно установить индивидуальные дополнения для каждого сервера. Иногда, в зависимости от предъявляемых требований, администратору может понадобиться несколько шаблонов для своих надобностей.

Относительно настроек сети — соединение, настроенное ранее, во время установки, работает как сетевой мост, давая возможность другим виртуальным машинам появиться в сети. Иными словами, виртуальные гости и устройства появятся в той же сети, что и сервер ProxMox.

Теперь пора загрузить ISO-образы желаемых виртуальных серверов, которые будут созданы. Чтобы загрузить установочные диски, выберите вид сервера в управляющей web-консоли и перейдите в локальный диск. Отобразятся все диски, так что щелкните по новому диску. Кнопка Upload [Загрузить] должна быть доступна на ленте меню над ними. Нажмите на Upload и выберите локальный ISO-образ в своей локальной системе. Системе понадобится несколько минут, чтобы это усвоить и стать доступной для хоста ProxMox.

Создание машин

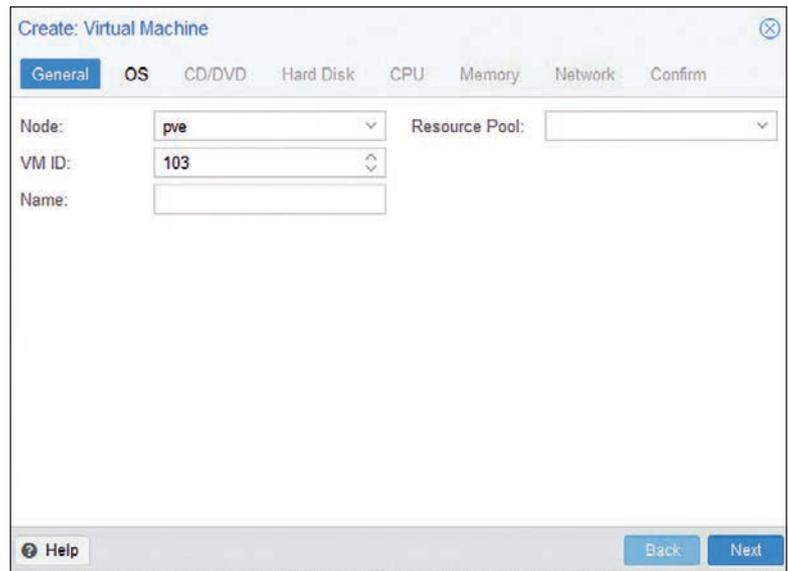
Чтобы создать виртуальную машину, убедитесь, что у вас включен вид Server, и нажмите Create VM в верхнем правом кластере кнопок. Откроется мастер создания VM. Каждой виртуальной машине присваивается уникальный номер (ID), начинающийся с 100. На этом первом экране дайте серверу имя (в поле имени), избегая использования пробелов. Оставьте прочие настройки как есть и нажмите Next [Далее].

Следующий экран покажет различные варианты Linux и Windows. В этом примере мы установим копию Ubuntu Linux Server, поэтому выберите Linux 4.x/3.x/2.6 Kernel. Это позволит системе использовать оптимальные настройки по умолчанию для версии ОС, а мастера по умолчанию заставит придерживаться оптимальных опций настройки в плане типов диска. В меню жесткого диска оставьте большинство опций по умолчанию. Единственное поле, которое в данном примере вы, возможно, захотите изменить, это объем диска. Нажмите Next. На вкладке CPU мы советуем использовать два сокета и оставить всё остальное по умолчанию.

С памятью всё несколько сложнее. По умолчанию администратор может выделить фиксированный объем ОЗУ. Однако при настройке ОЗУ можно быть чуть более гибким. Разрешается также настроить более гибкий минимум и максимум.

При появлении ядра 3.x поддержка гостевой KVM не была встроенной. Нажмите на кнопку-переключатель Linux 4.x, затем на Next. Используя ранее загруженный образ Ubuntu (по умолчанию выбран), выберите из выпадающего списка диск с ISO-образом. Выберите ISO-образ для использования (например, загруженный ранее) и нажмите Next, чтобы перейти на страницу настройки диска. На следующей странице для нашего простого примера оставьте сетевые настройки как есть. И далее подтвердите создание VM, нажав на Finish [Завершить].

На следующем экране настройте диск должным образом. Для нашей базовой настройки отлично подойдет выбор по умолчанию. На следующей странице мы бы рекомендовали настроить использование ОЗУ на 512 МБ, а лучше на 1024 МБ, чтобы VM при работе не испытывала нехватки памяти. Оставьте сеть по умолчанию и подтвердите настройку.



➤ **Настройка и установка VM проста: она сводится к использованию мастера настройки VM и прикрепленного файла ISO.**

В списке задач должен появиться новый пункт. По завершении нажмите на кнопку консоли (справа). Это даст доступ к консоли и позволит пользователю взаимодействовать с ней. В этот момент администратор должен быть в состоянии установить выбранную версию Linux. Большинство современных дистрибутивов будут работать без проблем по умолчанию, поскольку оптимизированные драйверы виртуализации встроены в ядро.

Чтобы начать установку Ubuntu на VM, перейдите во вновь созданную VM и нажмите Start на ленте кнопок в верхней правой части страницы. Открытие окна консоли позволит напрямую взаимодействовать с консолью VM. Эта установка идентична обычной установке Ubuntu. По завершении удаленное управление сервером должно быть не сложнее, чем использование SSH.

ProxMox также поддерживает ряд серверов Windows, если это нужно. К сожалению, драйверы виртуализации не встроены в Windows по умолчанию. Поэтому установка немного сложнее, и администратору во время установки потребуются установить драйверы, загружаемые при инициализации системы.

Наше пошаговое руководство затронуло только самое основное из того, что можно сделать с помощью ProxMox. Это простой в использовании, но также мощный и многофункциональный гипервизор. По мере привыкания администратора к ProxMox можно начать экспериментировать с лежащими в основе инструментами KVM и прочими функциями, например, с настройкой кластеров с помощью добавления хостов и хранения с распределенным доступом. Эти функции выводят ProxMox на новый уровень применения. Если администратор решит развивать свою настройку далее, более подробную информацию можно найти на сайтах и форумах ProxMox. www.linuxformat.ru/subscribe

Скорая помощь

Если на клиентской системе еще не установлен клиент SSH, знайте: в web-интерфейсе встроена виртуальная консоль. Просто выберите Shell из контекстного меню слева.

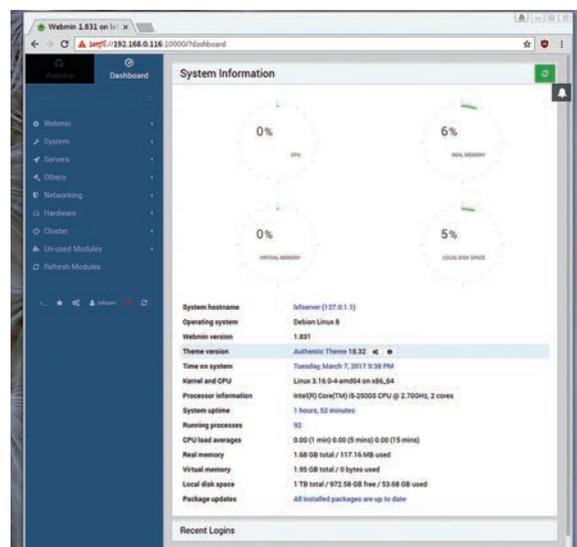
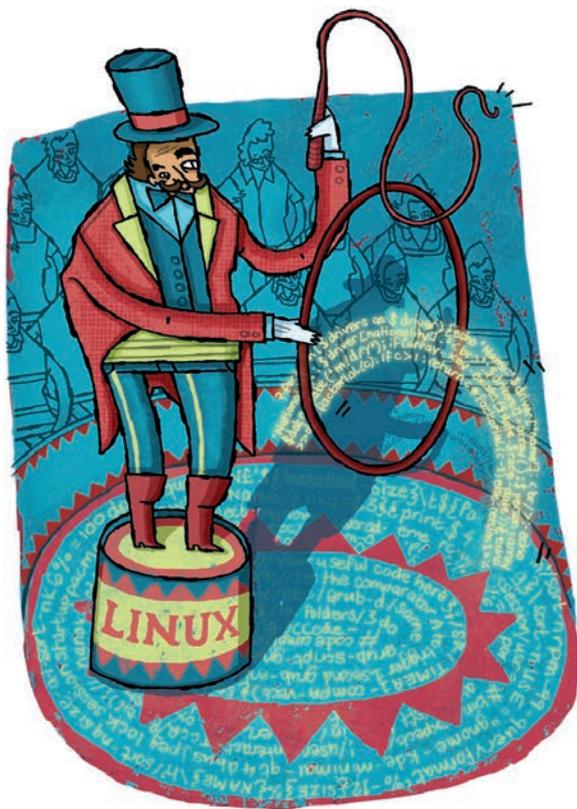
Сервер LXF: Время Samba

Можно ли написать учебник по Samba без танцевальной шутки? Джонни Бидвелл притопывает одной ногой, затем другой, и вот уже пляшет босса-нова.



Наш эксперт

Джонни Бидвелл рад, что пережил зиму. Дни становятся длиннее, птички трели наполняют вечерний воздух, и поблизости слышны звуки психоделического транса. Минуточку, последнее-то верно и зимой.



» Панель инструментов *Webmin* предоставляет полезный обзор состояния вашего сервера. Сейчас там мало что есть.

Скорая помощь

Если у вас в сети только Linux-машины, понятно, что вы предпочтете протокол NFS. Он не лишен причуд (в частности, когда серверные и клиентские версии не совпадают), поэтому не ждите более легкой прогулки. У NFS к тому же нет собственного модуля *Webmin*.

Если вы читали наш предыдущий урок, то видели, как можно получить доступ к файлам на сервере через SSH-службу, используя синтаксис `ftp://` в файловом менеджере. Для этого есть также средства командной строки, а именно `ftp` и `scp`, но в мире много операционных систем, и не все они будут охотно плясать под дудку SSH. Кроме того, отправка больших объемов данных через SSH довольно неэффективна — он никогда не предназначался для такой нелепости. У Linux есть свой механизм обмена файлами, но NFS, семейство протоколов SMB (Server Message Block) в Windows и в особенности диалект CIFS (Common Internet File System) сегодня стали для этого стандартом де-факто. Поскольку работа с файлами — одна из самых полезных функций, выполняемых сервером, было бы неплохо, если бы наша машина танцевала под музыку CIFS. И благодаря Samba, она может!

Прежде чем мы приступим, несколько стандартных предостережений. Во-первых, никогда не делитесь всей корневой файловой системой с доступом на запись. Не стоит. Во-вторых, также, как

правило, не очень правильно делиться чем-то личным, даже если это доступно только для чтения и даже если якобы доступно только членам семьи или партнерам. Если одно из их устройств будет взломано, то же, вероятно, произойдет и с вашими данными. Наконец, хотя наш урок ориентирован на IPv4 и большинство домашних сетей в целом скрыты от ужасов Интернета за слоем NAT, мир постепенно переходит на IPv6. В частности, британские интернет-провайдеры начинают предлагать подключение по IPv6, и при отсутствии подходящего брандмауэра, для машин, подключенных таким образом, нет понятия локальной сети — любая машина в мире может подключиться к их сервисам. Поэтому будьте осторожны в предоставлении сервисов по протоколу IPv6. А мы будем явно деактивировать его для Samba и *Webmin*.

С учетом всего этого мы можем обновить программное обеспечение нашего сервера и установить необходимые инструменты. Это делается либо путем входа в систему по старинке, либо по SSH из любого места вашей сети. Рекомендуется всегда обновлять пакеты на своем сервере, выполняя следующее:

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

Debian, как правило, предоставляет только обновления безопасности, так что эти команды не должны приводить к загрузке огромного объема данных и нарушать работу из-за капитального

Установка Samba и Webmin

С обновленной базой данных пакетов можно установить Samba и ее зависимости простой командой

```
$ sudo apt-get install samba
```

Если в этот момент вам предложат вставить установочный CD, у вас, как и раньше, есть два варианта. Сделайте то, что предлагается, или прокомментируйте в `/etc/apt/sources.list` строку, начинающуюся с `deb cdrom`. Debian, похоже, чрезмерно заботится здесь о пропускной способности, но поскольку пакеты на установочном носителе всё равно скоро устареют, это не слишком помогает в широком смысле.

Webmin сложнее в установке — его нет в официальных репозиториях Debian, но доступен файл `.deb`, который обеспечит установку зависимостей пакета, и он будет обновлен вместе с остальной системой во время стандартной процедуры.

Чтобы его использовать, сначала нужно добавить в *apt* репозиторий автора и ключ, таким образом:

```
$ sudo sh -c 'echo "deb http://download.webmin.com/download/repository sarge contrib" > /etc/apt/sources.list.d/webmin.list'
```

```
$ wget -qO - http://www.webmin.com/jcameron-key.asc | sudo apt-key add -
```

OK

Будьте внимательны с одиночными и двойными кавычками в первой команде, а во второй учтите, что после `-q` идет буква `O`, а не ноль (этот параметр вместе с последующим дефисом выводит загруженный контент в стандартный вывод, который у нас перенаправлен в утилиту добавления ключа).

Готово? Теперь обновите списки пакетов (снова) и установите *Webmin*:

```
$ sudo apt-get update
```

```
$ sudo apt-get install webmin
```

Вот и всё.

обновления версии. Конечно, если вы долго не обновляетесь, то повышается риск взлома, поэтому старайтесь делать это хотя бы еженедельно. Это также позволяет избежать ситуации, когда вы пытаетесь установить устаревшую версию пакета, которая была удалена из репозитория.

Готовимся к дележке

В прошлый раз мы говорили о создании большого раздела для хранения данных и его монтировании в `/mnt/data`. Но мы ничего не делали с этим каталогом, предоставив установщику всё это настроить. Если команда

```
$ grep /mnt/data /etc/fstab
```

ничего не возвращает, и вы не настроили свой накопитель или раздел явно, надо будет сделать это сейчас, поскольку этот каталог мы будем использовать как общий через Samba. Команда `lsblk` отобразит список дисков и их разделов. Вы не сможете работать с диска, содержащего ОС, хотя можно переместить и добавить необходимый раздел данных с живого носителя. Допустим, вы только что установили диск `/dev/sdb` и хотите весь его использовать как раздел данных. Сначала диск надо разбить, это делается по `sudo fdisk /dev/sdb`. Введите `p`, чтобы отобразить таблицу разделов. Если на диске есть другие разделы, проверьте, нет ли там чего важного. Предположим, что нет; тогда удалите их все, набрав `d`. Теперь создайте новый раздел, введя `n` и приняв значения по умолчанию. Затем запишите таблицу разделов, введя `w`.

Прежде чем раздел может быть использован, его надо отформатировать. Мы собираемся использовать файловую систему `ext4`, и это хороший выбор, если у вас нет особой причины взять нечто более экзотическое. Будьте особенно внимательны при вводе этой команды, потому что она удалит всё на целевом разделе, и вероятность восстановления ничтожна:

```
$ sudo mkfs.ext4 /dev/sdb1
```

Теперь надо велеть Debian монтировать эту файловую систему при каждой загрузке. Команда `sudo blkid` перечислит все блочные устройства с их UUID (преимущество UUID в том, что они никогда не меняются, даже если добавляются или удаляются другие диски, в отличие от обозначений устройств `sdb1` и т.д.). Следующий шаг будет намного проще, если вы зайдете на свой сервер по SSH из среды рабочего стола с возможностью вырезания и вставки [Ред.: — Новомодные игры с буфером обмена... в мое время такого не было]. Скопируйте, запишите или запомните соответствующий UUID. Затем отредактируйте таблицу файловой системы с помощью

```
$ sudo nano /etc/fstab
```

Добавьте строку вида

```
UUID="abcdef01-2345-6789-abcd-ef012345678" /mnt/data ext4 defaults 0 0
```

вставив этот UUID, сохраните и выйдите с помощью `Ctrl+X, Y, Enter`. Теперь легко смонтировать наш раздел данных:

```
$ sudo mount /mnt/data
```

Давайте создадим в этом каталоге папку `incoming`. Дадим этому каталогу достаточно широкие права доступа, так как мы хотим сделать его доступным для записи гостям через Samba, чтобы любой мог что-то загрузить (но также и удалить).

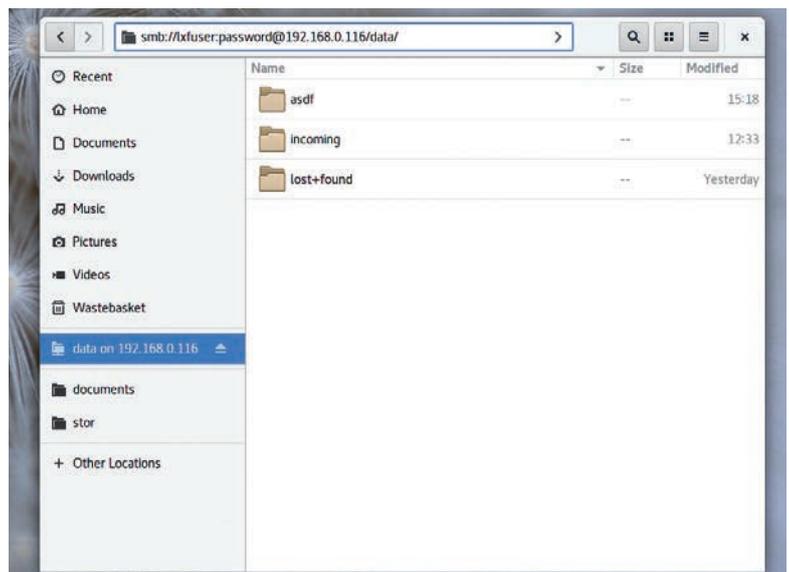
```
$ sudo mkdir /mnt/data/incoming
```

```
$ sudo chmod 777 /mnt/data/incoming
```

Заодно изменим владельца `/mnt/data`, что даст привилегированным пользователям доступ к записи в этот каталог и в `incoming`.

Наша следующая задача — установить героев этой истории, Samba и *Webmin*. Подробно узнать, как это делается, можно в верхней части страницы. После этого мы увидим сообщение, что теперь можно войти в *Webmin* по адресу <https://lxfserver:10000/>. Строго говоря, это неправда, поскольку остальная часть нашей сети не знает имени хоста сервера. Не будем ворошить муравейник, связанный с именем хоста; лучше притворимся, что сообщение фактически сказало нам, что теперь мы можем зайти на <https://192.168.1.100:10000/>, помня, что это IP-адрес, который

»



» Gnome Files и почти все современные файловые менеджеры предоставляют доступ к своим ресурсам Samba в таком синтаксисе. Кстати, это не наш пароль.

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

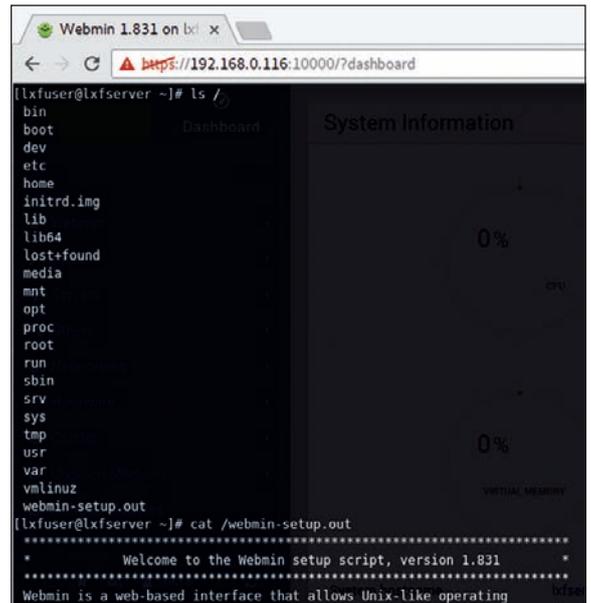


Сamba была первоначально написана Эндрю «Триджеллом» Триджеллом, и часто утверждают, что его работа была связана с обратным инжинирингом. Триджелл не совсем согласен с этим термином, и в 2003 г. предложил аналогию с «French cafe», когда, не зная французского языка, вы слушаете, какие слова посетителей приводят к желаемому для вас результату, и ухитряетесь сделать заказ официанту. Полностью см. на <http://bit.ly/2myDKqG>.

мы назначили нашему серверу в прошлый раз (явно измените его в соответствии с вашей конфигурацией). Но не всё так просто (а когда было просто?), поскольку при попытке посетить эту страницу вы получите предупреждение о недопустимом сертификате. Хотя обычно игнорировать его было бы неправильно, в данном случае это нормально, поскольку мы уже неявно согласились, что это адрес нашего сервера, когда зашли по SSH для установки всех этих битов раньше. Web-браузеры затрудняют обход этого препятствия, но способ всегда найдется, если не полениться прочитать то, что написано мелким шрифтом. В *Chrome/Chromium* следует выбрать **Advanced > Proceed to 192.168.1.100 (unsafe)** [Дополнительно > Проследовать с 192.168.1.100 (небезопасно)]. В *Firefox* всё похоже: **Advanced > Add Exception > Confirm Security Exception** [Дополнительно > Добавить исключение > Подтвердить исключение в защите] с опцией постоянного сохранения для этого сайта, чтобы больше не мешало.

В *Webmin* есть что изучать — некоторые важные моменты можно найти во врезке на стр. 65, но давайте сначала сосредоточимся на нашей задаче. Настройка Samba выполняется в секции **Servers** в левой колонке раздела **Samba Windows File Sharing**. Выберите в верхней строке **Create A New File Share**. В поле **Share name** введите **data**, а в поле каталога ниже введите **/mnt/data**. Откажитесь от предложения «Automatically create directory [Создать директорию автоматически]» ниже, поскольку мы только что сделали это сами, и убедитесь, что **Available**, и **Browseable** включены. В поле **Share Comment** введите что-нибудь информативное или забавное, а затем нажмите **Create**.

Теперь ваш общий каталог должен быть виден в сети, что можно проверить, открыв файловый менеджер, нажав **Ctrl+L** и введя **smb://192.168.1.100**. После этого вы сможете подключиться к общему ресурсу как **Guest** (пароль не требуется) и просмотреть там каталог **incoming/**. Внутри каталога **data** можно увидеть папку **lost-found**, которая создается новой файловой системой и используется для хранения фрагментов файлов, утерянных при некорректном завершении работы. О ней можете не беспокоиться. Чего

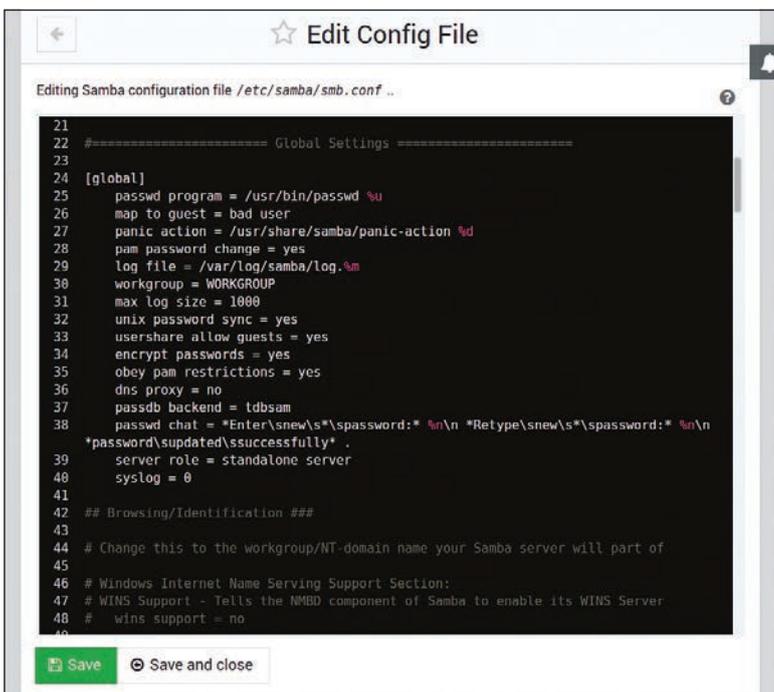


➤ Не одобряете новомодные web-шутки? *Webmin* позволяет делать это по-старому, в терминале, в браузере.

мы не сможем сделать, даже если нам этого хочется — создавать каталоги или помещать файлы в наш каталог **incoming/**.

Дело в том, что у Samba собственные механизмы прав доступа и фактически собственный каталог пользователей, и они работают независимо от «родных» конструкций Linux. Чтобы записывать внутри общего ресурса Samba, этот ресурс должен быть явно помечен как доступный для записи. Как вы и понадеялись, Samba тоже уважает права на каталог в Linux, так что их надо должным образом установить (как мы и поступили). Вернитесь к общему ресурсу **data** в *Webmin* и перейдите в меню **Security and Access Control**. Активируйте первую опцию, **Writeable**, и нажмите **Save**. Возможно, вам придется отключиться и снова подключиться к общему ресурсу, чтобы это сработало (у Gnome есть кнопка отключения рядом с общим ресурсом в левом столбце), но теперь вы можете без проблем писать внутри каталога **incoming**. Проверьте это, создав папку или скопировав файл из файлового менеджера. Теперь попробуйте сделать это из самого общего каталога. Это не работает, или, по крайней мере, не должно работать. Несмотря на то, что общий ресурс задан как доступный для записи, базовые разрешения каталога запрещают пользователю Samba по умолчанию (который соответствует учетной записи «nobody» в системе) там писать, отсюда и ошибка. Нам не надо, чтобы какой-нибудь старина Томас, Ричард или Гарольд могли писать (и особенно удалять) здесь содержимое, так что это хорошо. Однако было бы неплохо входить в систему с негостевой учетной записью, чтобы заполнить этот общий ресурс.

Именно здесь *Webmin* действительно блещет. Как мы уже упоминали, пользователи Samba отличаются от обычных учетных записей пользователей в системе. Можно определить новых пользователей и соответствия между двумя наборами, покорпвав над текстовыми файлами, но проще позволить *Webmin* сделать этот пустячок за нас. В разделе **Samba** в *Webmin* выберите **Convert Users**. Для простоты мы просто собираемся повторить нашего **lxfsuser** на стороне Samba, поэтому выберите верхний бокс (**Only listed users...**) и введите **lxfsuser** в соседнем поле. Щелкните **Convert Users** и вернитесь в общий список. Теперь в разделе **Samba Users** должен быть указан **lxfsuser**. Наша работа не завершена, поскольку новые версии Windows и Samba по умолчанию (разумно) требуют



➤ Видите, сколько боли, страданий и на первый взгляд повторяющихся параметров позволяет избежать *Webmin*, редактируя **smb.conf** в браузере.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Что сделать в Webmin (чтобы ваш сервер не упал)

Помимо снятия боли и страданий от Samba (и будет еще мучительнее, когда вы начнете получать неожиданные сопоставления пользователей и **smb.conf**), *Webmin* умеет администрировать практически все другие аспекты вашей системы. От заданий *cron* до пакетов программ и регистрации файлов пользователей, *Webmin* может, соответственно, редактировать, устанавливать, просматривать и запрещать всё это с комфортом web-браузера. Ранее для защиты нашего SSH-сервера от входа ботов мы установили *fail2ban*, и модуль для управления им есть в *Webmin* в разделе *Networking*. Кроме того,

вы можете настроить сам демон SSH в разделе *Servers*. Однако всё это удобство сопряжено с потенциальными потерями, а именно: каждый, кто имеет доступ к нашему имени пользователя, паролю и странице входа в *Webmin*, по существу имеет полный контроль над нашей системой. Это может быть не так плохо, как кажется — страница входа, вероятно, вне вашей домашней сети невидима. И всё же надо принять меры для снижения риска.

В разделе *Webmin* выберите *Webmin Configuration* > *IP Access control*. Мы ограничим доступ только локальным хостом и локальной сетью, введя в поле

127.0.0.1

192.168.1.0/24

Обязательно измените вторую строку в соответствии с адресом вашей сети, иначе вы больше не сможете получить доступ к *Webmin*. Сохраните эти параметры, затем вернитесь в раздел конфигурации, на сей раз открыв *Ports and Addresses*. Уберите флажок *Accept IPv6*, на случай, если в один отнюдь не прекрасный день кто-то посторонний изловчится сюда пробраться.

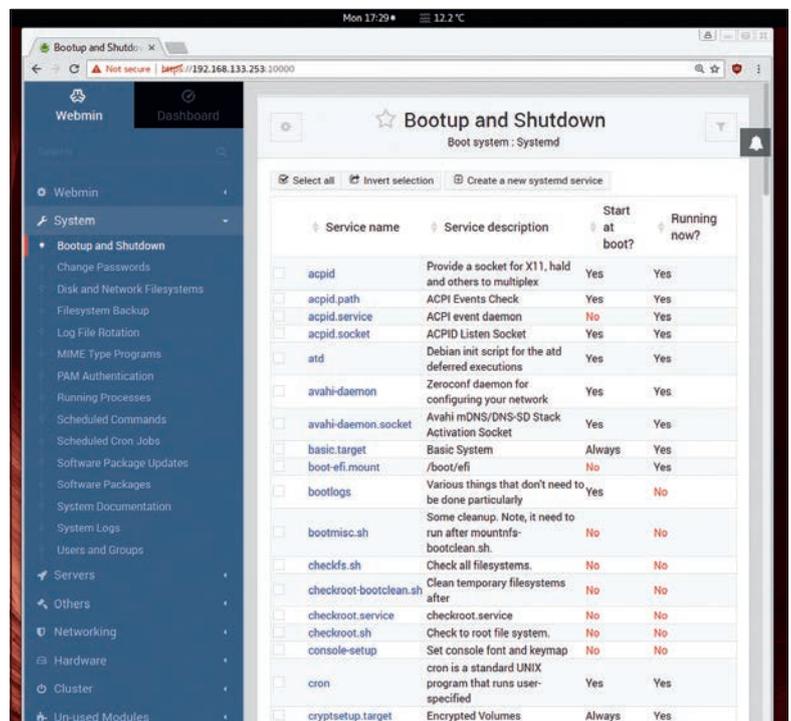
Ну вот, элементарные меры безопасности нами приняты.

шифрования паролей, поэтому наш текущий пароль (или даже любой пароль) нельзя автоматически перенести на пользователя Samba. Итак, щелкните по "lxfuser", переместите переключатель в *New Password*, введите нужный пароль и нажмите *Save*.

Для проверки новой учетной записи необходимо повторно подключиться с ней к общему ресурсу. Теоретически здесь годится синтаксис `smb://lxfuser@192.168.1.100`, который должен запросить пароль. Однако, похоже (по крайней мере, в Gnome), иногда бывает трудно входить в систему с другими учетными данными, и не исключено, что вам придется выйти из системы и снова войти в рабочий стол, чтобы проверить это. При удачном возврате в диалоговое окно входа в систему, выберите *Registered User* и используйте "lxfuser" с соответствующим паролем. Теперь вы будете в состоянии создавать файлы и каталоги в общем ресурсе *data/*, а также в *incoming/*.

Windows 10 груба с гостями

Наш вновь созданный сетевой ресурс должен быть видим клиентами, подключающимися с любых устройств. Если он не отображается в Windows, вы можете использовать нотацию `\\192.168.1.100` для подключения из *Explorer*. Однако в Windows 10 (или, по крайней мере, в ее последних сборках) вы обнаружите, что не можете подключиться в качестве гостя, поскольку по соображениям безопасности эта функция отключена. В домашней сети это не проблема, но аргументация Microsoft такова, что машины, пытающиеся автоматически подключиться к этим ресурсам, в других сетях могут быть обмануты и подключиться к враждебным серверам. Как мы видели, не особенно сложно защитить ваш ресурс паролем, и если подходить к этому вопросу строго, мы должны так делать со всеми. Но удобство входа без пароля тоже неплохо, и можно его



► Не понимаете или не желаете понимать, как управлять службами *systemd*? Есть апплет *Samba*, который делает это через флажки и кнопки.

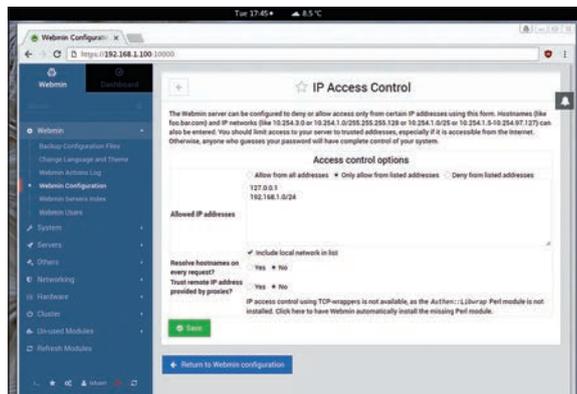
вернуть, добавив ключ реестра на машине Windows — подробнее см. <http://bit.ly/1Y44qeP>.

Что касается безопасности, то действительно необходимо остановить прослушивания интерфейса IPv6 в Samba, добавив следующие строки в `/etc/smb.conf`:

```
interfaces = 192.168.1.100
```

```
bind interfaces only = Yes
```

И это более или менее завершает тему нашего урока; следите за появлением очередной захватывающей главы в нашей серии о серверах. Но если вам невтерпёж ждать, рассмотрите возможность добавления собственных служб на сервер. Из *Webmin* можно сделать многое, но ограничиваться им не стоит. Следуя рецепту нашего урока, легко настроить общий ресурс Samba, чтобы делиться музыкой через вашу сеть; но столь же легко настроить *Emby* для намного более эффективной трансляции и через удобный web-интерфейс. Или даже можно использовать MPD (*Music Player Daemon*), чтобы настроить ваш сервер как музыкальный проигрыватель с управлением через сеть. Просто добавьте большие громкоговорители, мощный усилитель и несколько гигабайт психотранса. LXF



► Ограничение доступа к *Webmin*, вероятно, шаг разумный. Если вы заблокируете сами себя, редактирование файла настройки исправит это.

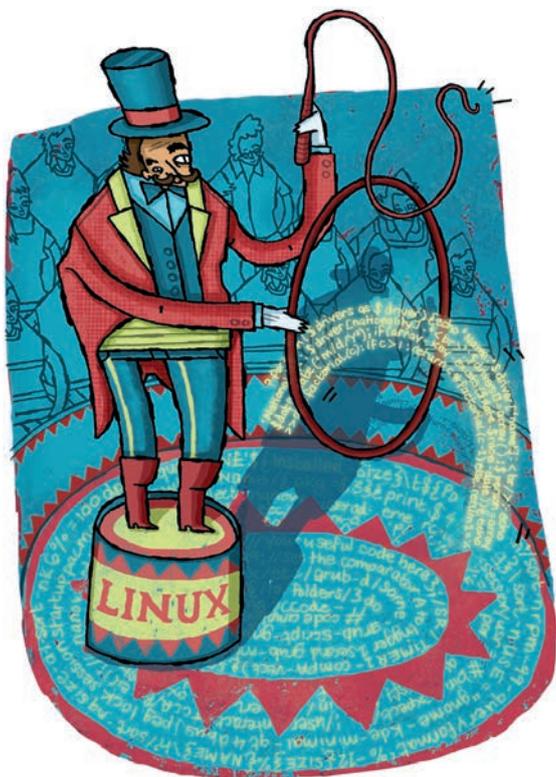
GnuPG: Рудим своими ключами

Ваш ключ GnuPG — ваше сокровище. Джон Лэйн расскажет, как создать надежный ключ и уберечь его от онлайн-воришек.



Наш эксперт

Джон Лэйн воспользовался бы GnuPG, если бы знал кого-нибудь, кто так делает. А пока он будет просто шифровать свой ключ.



GnuPG, он же Gnu Privacy Guard — это инструмент для защиты личной информации. С его помощью можно зашифровать электронную переписку или проверить подлинность программных пакетов перед их установкой. В нем реализован стандарт OpenPGP, в основе которого лежит шифрование с применением открытых ключей, подтверждаемых посредством сертификатов и личного доверия.

В начале документации GnuPG говорится о том, как важно правильно управлять ключами. Именно это станет предметом нашего урока, в котором мы разберем, как это лучше осуществить. Мы будем использовать GnuPG версии 2.1, по прозвищу Modern.

Некоторые основы

Ключ GnuPG имеет открытые и закрытые компоненты, которые по отдельности называются открытым (публичным) и закрытым (приватным) ключом, а вместе — ключевой парой. Закрытые ключи используются для подписания и дешифровки, а открытые — для проверки подписи и шифрования. Открытые ключи хранятся в «сертификатах» вместе с идентифицирующей информацией (идентификатором пользователя, которым могут быть имя, адрес электронной почты или, возможно, фотография) плюс одним



► SKS Key Servers — один из ряда открытых серверов управления ключами; через его интерактивный онлайн-интерфейс вы можете загрузить свой ключ или найти те, что принадлежат другим людям.

или несколькими связанными «подключками», позволяющими использовать отдельные ключи для подписания и шифрования. Существование подключей объясняется историей разработки GnuPG и актуальными проблемами безопасности.

Пользователь GnuPG собирает открытые ключи других людей, возможно, получив от сервера ключей сертификат, а также собственную пару ключей GnuPG. В зависимости от контекста «ключ» может означать пару ключей, закрытый ключ, открытый ключ, подключ или сертификат.

Мы рассмотрим, как всё это условно выглядит для нового пользователя, которому не терпится всё зашифровать. Давайте создадим этого пользователя, как-то так:

```
$ sudo useradd -m alice
$ sudo -i -u alice
```

Задайте настройки

Первым делом надо настроить GnuPG. Он использует каталог (по умолчанию — `~/gnupg`) для хранения своей конфигурации, данных о времени выполнения и связке ключей. Он может создать этот каталог автоматически, но если вы сделаете это вручную, то перед первым использованием GnuPG сможете задать некоторые настройки. Сделайте это, используя права с ограниченным доступом:

```
$ mkdir -m 700 ~/gnupg
```

Файл настройки создавать не обязательно (GnuPG предполагает конфигурацию по умолчанию), но написание этого файла позволит нам установить некоторые индивидуальные параметры:

```
# ~/gnupg/gpg.conf
keyserver hks://hks.pool.sks-keyservers.net
personal-cipher-preferences AES256 AES192 AES CAST5
personal-digest-preferences SHA512 SHA384 SHA256 SHA224
cert-digest-algo SHA512
```

Скорая помощь

Если при создании нового ключа вы хотите выбрать его тип, алгоритм шифрования и размер ключа интерактивно, используйте `gpg --full-genkey`.

```
default-preference-list SHA512 SHA384 SHA256 SHA22 AES256
AES192 AES CAST5 ZLIB BZIP2 ZIP Uncompressed
```

В этом примере конфигурации создается сервер ключей по умолчанию; от него можно получить сертификаты открытых ключей, а также предпочтительные алгоритмы шифрования и пересчета (хэширования). В список настроек по умолчанию входят те, что будут применяться при создании новых ключей, дабы используя их третьи стороны знали о наших предпочтениях. В совокупности эти предпочтения определяют, какие алгоритмы сможет использовать GnuPG, и мы, для примера, выберем самые надежные. Однако алгоритмы по умолчанию, если вы решите придерживаться их, для большинства случаев подходят.

Простейший способ создать ключ — набрать `gpg --gen-key` и следовать подсказкам, которые запрашивают имя, адрес электронной почты и парольную фразу. Этот метод использует настройки по умолчанию (в том числе из файла конфигурации) и будет выдавать незашифрованный 2048-битный «первичный» или «главный» ключ RSA для подписания и сертификации, идентификатор пользователя, составленный из имени и электронной почты, и подключ (тоже 2048-битный RSA) для шифрования.

Другой подход состоит в использовании пакетного режима GnuPG, он позволяет получить требуемую информацию из файла параметров, без интерактивного ввода. Файл параметров обеспечивает более детальную настройку, чем интерактивный инструмент, а также служит для записи параметров, используемых для генерации ключа.

```
$ cat <<EOF > alice.keyparams
```

```
Key-Type: RSA
```

```
Key-Length: 4096
```

```
Key-Usage: sign
```

```
Subkey-Type: RSA
```

```
Subkey-Length: 4096
```

```
Subkey-Usage: encrypt
```

```
Name-Real: Alice
```

```
Name-Email: alice@example.org
```

```
Passphrase: alice1234
```

```
Expire-Date: 1y
```

```
EOF
```

```
$ gpg --verbose --gen-key --batch alice.keyparams
```

Генерация ключа может занять некоторое время, поскольку для получения достаточных случайных данных требуется накопить «энтропию»; опция `verbose` велит `gpg` выдавать обратную связь о ходе выполнения.

Наш пример генерирует большие 4096-битные ключи (просто для иллюстрации возможностей, хотя в большинстве случаев подойдут 2048-битные ключи), срок действия которых истекает через год после создания. Ключ с истекшим сроком действия недействителен и не может применяться для подписи или шифрования, а установка даты, когда это произойдет, является мерой предосторожности, нетрудной, но полезной в том случае, если личный ключ или его кодовая фраза потеряются. Тогда, даже если ключ нельзя отменить, вас утешит то, что в итоге он устаревает.

А если ничего плохого не случится, можно самим изменить дату окончания срока действия, даже если ключ уже устарел. Учтите, что просроченные ключи всё равно могут дешифровать ранее зашифрованные сообщения.

Приведенный пример файла параметров включает кодовую фразу, но если в файле она не указана, `gpg` запросит ее интерактивно. Помимо этого, можно задать «глобальные параметры», которые определяют список настроек по умолчанию в `gpg.conf`.

Идентификатор пользователя состоит из имени (Name-Real), электронной почты (Name-Email) и необязательного комментария (Name-Comment), который мы не ввели. В сообществе PGP рекомендуют не использовать комментарии, поскольку они не являются идентифицирующей информацией. Изменить идентификатор пользователя нельзя, но можно отозвать его и добавить новый.

Ключ в том, что...

По завершении генерации ключей `gpg` может отобразить отчет о выполненных действиях, по команде с `--list-keys` (или `-k`):

```
$ gpg -k alice
```

```
pub rsa4096 2016-10-03 [SC] [expires: 2017-10-03]
```

```
109FB60CAD48C7820CF441A661EB6F7F34CE2E54
```

```
uid [ultimate] Alice <alice@example.org>
```

```
sub rsa4096 2016-10-03 [E] [expires: 2017-10-03]
```

Здесь мы видим три вещи: 4096-битный первичный ключ, идентификатор пользователя и подключ. Пометка `[ultimate]` у идентификатора пользователя выражает полное доверие (ключ-то ваш) и то, что ключ действителен и вы будете доверять всем другим ключам, которые подписываете с его помощью.

Длинная строка шестнадцатеричных символов является «отпечатком» первичного ключа, 160-битным SHA1-хэшем его содержимого. Префикс `pub` первичного ключа сообщает, что это открытый ключ; префикс `sub` передает аналогичное значение подключа. Два соответствующих закрытых ключа не перечислены, но также существуют во вновь создаваемой связке ключей (которая хранится в каталоге `~/.gnupg`). Чтобы посмотреть все, скомандуйте `gpg --list-secret-keys` (или с краткой формой параметра — `-K`), см. выше:

```
# Each base16 line ends with a CRC-24 of that line.
# The entire block of data ends with a CRC-24 of the entire block of data.
1: 00 04 58 E1 AD EB 6D FB 8F 9F 19 B5 58 6F 50 D8 F0 AC C1 4C CC 8F B51363
2: 02 B9 FE 07 03 02 3C 61 CE EC 6A 4B E6 A4 D0 A4 AE F1 4F 52 27 EE 2AA7E2
3: AB A6 1A 66 0D 20 8C 94 14 C8 8F 72 5E EC 7B E4 78 CA 8F 69 4C A8 73932D
4: 80 7B 7F 54 8D 25 21 92 2F C6 91 01 9D A5 C6 B6 28 FC 3F 35 42 85 961B5E
5: 4B CE DB 6F 99 F2 7F 95 CC 35 24 8C 40 4E 3D C1 1F 48 86 AD 95 AB 4822BC
6: BA 68 9D 0A B8 FB F9 E1 39 08 88 77 A1 C0 7C 0E 4C 9C 08 EF 9E 66 56FF4E
7: 6E 3D 18 D1 C7 E4 80 00 68 4E 57 94 DD 9B A5 F4 14 35 8D 57 8D DB 384123
8: 91 EB D9 65 DA 4D 91 06 08 65 00 72 4B EC 19 E8 46 B2 F8 2A E5 D7 19E80F
9: 48 8F F8 DA 13 57 13 FD DA 40 43 E1 AA 5C 04 C6 77 5E AA EC 6F F9 A0843E
10: CF 8A 03 63 56 7E B5 78 D5 23 31 AD FF 3C AF 7C 7E CE 4E 74 6F 1B D75060
11: EF C7 CD 93 11 FA 25 02 FC 2C 64 51 CE E8 F5 EA 13 10 B1 92 3A 57 0AD2DD
12: 9F D7 2E 18 66 61 31 0D DA B8 43 A2 8E 43 D6 29 18 56 95 A5 E7 F8 518912
13: 79 8D BE E1 58 54 80 5C 79 0A 75 EC 22 87 BA 15 DC 2C 98 C3 9B 8F C778B6
14: CE EC 88 57 B8 A9 7D 46 89 E2 DA B1 E0 54 8A C4 16 67 5F 7B C7 D7 FA35F9
15: 86 1E D7 F5 87 E8 4C 9E 9B 7B 98 18 43 E8 D6 8F 89 8E AA 17 AD FB 6ADADE
16: 0B AC A0 2D 4A 13 43 A7 74 1E FA 33 39 12 BC E5 75 C8 D8 90 A1 22 D5765C
17: 05 03 04 74 16 05 08 76 56 A0 A7 56 00 03 46 05 F1 C4 0B 66 35 09 0133A
```

➤ Для восстановления вашего открытого ключа можно употребить вывод *Paperkey*.

Скорая
помощь



Утилита *haveged* (www.issihosts.com/haveged) может обеспечить энтропию, необходимую для генерации ключей. Поищите ее в репозитории пакетов вашего дистрибутива.

Скорая
помощь



Вы можете добавить часто используемые параметры форматирования в `gpg.conf`. Просто уберите двойной дефис в начале файла.

keygen, pinentry, su, sudo и tty

GnuPG для управления ключами `gpg` использует демон `gpg-agent`, а в нем применяется вспомогательный инструмент `pinentry`, когда требуется ввод интерактивных кодовых слов. Для правильной работы `tty`-устройство вашей оболочки обязано принадлежать вам (например, для Alice `stat -c %U $(tty)` должно быть `alice`). Но при обращении через `sudo` или `su` могут возникнуть проблемы с такими

задачами, как генерация ключей, которые требуют доступа к секретным ключам. Может появиться сообщение об ошибке:

```
gpg: agent_genkey failed: Permission denied
```

Если это случится, попробуйте использовать маленький трюк, сгенерировав ключ в принадлежащем вам `tty` командой `script`:

```
$ script -q -c "gpg ..." /dev/null
```

где команда, которую вы хотите использовать, плюс все ее аргументы указаны в кавычках. Кроме того, вы можете воспользоваться `gpg --pinentry-mode loopback`, чтобы вместо диалога `pinentry` работать с командной строкой. Или, перед `sudo` или `su`:

```
$ sudo chown alice $(tty)
```

Может также потребоваться экспорт — `export GPG_TTY=$(tty)`.

Скорая помощь



Отыщите в своем репозитории либо в Интернете *paperkey*, *ssss* или *libgfs*.

```
$ gpg -K
sec rsa4096 2016-10-03 [SC] [expires: 2017-10-03]
109FB60CAD48C7820CF441A661EB6F7F34CE2E54
uid [ultimate] Alice <alice@example.org>
ssb rsa4096 2016-10-03 [E] [expires: 2017-10-03]
```

Здесь мы видим аналогичный вывод, только с префиксом `sec` (`secret`) для первичного закрытого ключа и `ssb` для подключа.

Основной способ идентифицировать ключ — это его отпечаток, но таковые обычно используются в сокращенной форме, состоящей из последних восьми символов. Короткие идентификаторы ключей способны конфликтовать, и вместо них лучше использовать более длинный идентификатор ключа, в 64 бита (последние 16 символов). Они тоже подвержены конфликтам, но в меньшей степени; если сомневаетесь, используйте полный отпечаток!

Добавьте к `gpg` параметры `--keyid-format short`, чтобы запросить формат короткого ключа, или `--keyid-format long` для длинного. Если набрать `--fingerprint`, отпечаток будет отображаться в более удобочитаемой форме. Вот так:

```
> short "34CE2E54"
> long "61EB6F7F34CE2E54"
> fingerprint "109F B60C AD48 C782 0CF4 41A6 61EB 6F7F 34CE 2E54"
```

Отпечаток подключа не отображается, если `--fingerprint` не упомянуто дважды.

Наш новый ключ «самоподтвержден»: это значит, что первичный ключ подписывает идентификатор пользователя и подключа, подтверждая, что он ими владеет. Можно просмотреть эти подписи:

```
$ gpg --list-sigs alice
pub rsa4096 2016-10-03 [SC] [expires: 2017-10-03]
uid [ultimate] Alice <alice@example.org>
sig 3 61EB6F7F34CE2E54 2016-10-03 Alice <alice@example.org>
sub rsa4096 2016-10-03 [E] [expires: 2017-10-03]
sig 61EB6F7F34CE2E54 2016-10-03 Alice <alice@example.org>
```

Подписи отображаются после того, к чему они относятся. На дисплее отображаются две подписи: одна для идентификатора пользователя, другая для раздела.

Две строки «sig», приведенные выше, отображают идентификатор ключа подписи, дату и идентификатор пользователя. Пробел после `sig` может содержать флажки, отображаемые на подписи первичного ключа, например, `3`; флажки определяют степень тщательности проверки ключа. Для подписей, создаваемых во время генерации ключа, GnuPG назначает наивысший уровень, `3`. В дополнение к подписям самосертификации, добавленным GnuPG, третьи лица могут также подписать записи `uid` на вашем ключе, чтобы подтвердить свою уверенность в том, что ключ и идентификатор пользователя принадлежат друг другу и вам. Такие подписи укрепляют ваш ключ и помогают вам создать свою сеть доверия.

Итак, базовый ключ создан, подписан и готов к работе. Вы можете использовать его как есть, но подумайте, что будет, если он потеряется: кто-то другой сможет с его помощью подписывать документы от вашего имени и читать зашифрованные документы, предназначенные только вам. Этот риск возрастет, если вы устанавливаете свой ключ на многих устройствах, особенно таких, как

ноутбуки, телефоны или планшеты, которые вы часто берете с собой в общественные места, где их легко украсть, забыть или утратить иным образом.

Ключ к совершенству

Прежде всего, на таких устройствах свой ключ лучше не устанавливать. Однако это не выход, если на них вам тоже надо подписывать и шифровать. Между тем, риск можно снизить с помощью подключаей, ведь у них есть удачное свойство: они могут быть отделены от первичного ключа, оставляя его дома, защищенным и оффлайн, и они могут устаревать или отменяться независимо.

Вы можете создать дополнительный подключа только для подписи, чтобы у вас их было два: один для подписи и другой для шифрования. Затем вы отсоединяете эти подключа от главного и устанавливаете только на своих носимых устройствах. Вот так:

```
$ gpg --edit-key alice
gpg> addkey
```

Интерактивный интерфейс требует определить тип ключа, его назначение (т.е. для подписи или для шифрования), длину (в битах) и длительность (срок действия), которая может быть бессрочной или указываться в днях, неделях, месяцах или годах.

При желании можно установить для подключаей отдельный пароль. Это делается либо в режиме редактирования, либо напрямую:

```
$ gpg --passwd alice
```

Независимо от выбранного метода, GnuPG выполняет итерацию по всем ключам, но вы можете отклонить те, которые не хотите изменять; указать один ключ невозможно.

Если у вас есть подключа, вы должны экспортировать их без первичного ключа:

```
$ gpg --export-secret-subkeys alice > subkeys.gpg
```

Перед экспортом подключаей GnuPG запросит пароль. Затем вы сможете посмотреть, что было экспортировано, используя временную связку ключей:

```
$ mkdir -m 700 .gnupg-temp
$ gpg --homedir .gnupg-temp --import subkeys.gpg
$ gpg --homedir .gnupg-temp -K
sec# rsa4096 2016-10-04 [SC] [expires: 2017-10-04]
ssb rsa4096 2016-10-04 [E] [expires: 2017-10-04]
ssb rsa2048 2016-10-04 [S] [expires: 2017-04-02]
```

Параметр `--homedir` велит GnuPG использовать вместо `~/gnupg` конкретный каталог (который может иметь любое имя, но должен быть уже существующим). Хэш-метка, следующая за тегом `sec`, указывает, что секретный ключ отсутствует в этой связке; присутствуют только секретные подключа. Аннотации показывают, что подключа могут шифровать (E) и подписывать (S) данные, и что первичный ключ может как подписывать данные, так и удостоверить (C) другие ключи.

Вы можете удалить временную связку ключей, как только прекратите ею пользоваться. Остановите демон и удалите каталог:

```
$ gpg-connect-agent --homedir .gnupg-temp KILLAGENT /bye
$ rm -r .gnupg-temp
```

Создание временной связки — это простой и легкий способ провести требуемую работу со своими ключами.

Скорая помощь



С помощью `gpg --list-packets` вы можете заглянуть внутрь своей связки ключей или файла экспорта и просмотреть внутренние данные OpenPGP; они задокументированы в формате **RFC4880**: (tools.ietf.org/html/rfc4880).

Подписи отображаются после того, к чему они относятся. На дисплее отображаются две подписи: одна для идентификатора пользователя, другая для раздела.

Две строки «sig», приведенные выше, отображают идентификатор ключа подписи, дату и идентификатор пользователя. Пробел после `sig` может содержать флажки, отображаемые на подписи первичного ключа, например, `3`; флажки определяют степень тщательности проверки ключа. Для подписей, создаваемых во время генерации ключа, GnuPG назначает наивысший уровень, `3`. В дополнение к подписям самосертификации, добавленным GnuPG, третьи лица могут также подписать записи `uid` на вашем ключе, чтобы подтвердить свою уверенность в том, что ключ и идентификатор пользователя принадлежат друг другу и вам. Такие подписи укрепляют ваш ключ и помогают вам создать свою сеть доверия.

Итак, базовый ключ создан, подписан и готов к работе. Вы можете использовать его как есть, но подумайте, что будет, если он потеряется: кто-то другой сможет с его помощью подписывать документы от вашего имени и читать зашифрованные документы, предназначенные только вам. Этот риск возрастет, если вы устанавливаете свой ключ на многих устройствах, особенно таких, как

ноутбуки, телефоны или планшеты, которые вы часто берете с собой в общественные места, где их легко украсть, забыть или утратить иным образом.

Прежде всего, на таких устройствах свой ключ лучше не устанавливать. Однако это не выход, если на них вам тоже надо подписывать и шифровать. Между тем, риск можно снизить с помощью подключаей, ведь у них есть удачное свойство: они могут быть отделены от первичного ключа, оставляя его дома, защищенным и оффлайн, и они могут устаревать или отменяться независимо.

Вы можете создать дополнительный подключа только для подписи, чтобы у вас их было два: один для подписи и другой для шифрования. Затем вы отсоединяете эти подключа от главного и устанавливаете только на своих носимых устройствах. Вот так:

```
$ gpg --edit-key alice
gpg> addkey
```

Интерактивный интерфейс требует определить тип ключа, его назначение (т.е. для подписи или для шифрования), длину (в битах) и длительность (срок действия), которая может быть бессрочной или указываться в днях, неделях, месяцах или годах.

При желании можно установить для подключаей отдельный пароль. Это делается либо в режиме редактирования, либо напрямую:

```
$ gpg --passwd alice
```

Независимо от выбранного метода, GnuPG выполняет итерацию по всем ключам, но вы можете отклонить те, которые не хотите изменять; указать один ключ невозможно.

Если у вас есть подключа, вы должны экспортировать их без первичного ключа:

```
$ gpg --export-secret-subkeys alice > subkeys.gpg
```

Перед экспортом подключаей GnuPG запросит пароль. Затем вы сможете посмотреть, что было экспортировано, используя временную связку ключей:

```
$ mkdir -m 700 .gnupg-temp
$ gpg --homedir .gnupg-temp --import subkeys.gpg
$ gpg --homedir .gnupg-temp -K
sec# rsa4096 2016-10-04 [SC] [expires: 2017-10-04]
ssb rsa4096 2016-10-04 [E] [expires: 2017-10-04]
ssb rsa2048 2016-10-04 [S] [expires: 2017-04-02]
```

Перекрестная сертификация

Есть риск, что открытый подключа может быть присоединен к другому сертификату, владелец которого затем может утверждать, что подписал документ. Чтобы предотвратить такой сценарий, GnuPG теперь проверяет, что ключи-подписи перекрестно сертифицированы, прежде чем проверять подписи. Кросс-сертификация требует, чтобы подключа

подписывали первичный ключ, с целью подтвердить их подлинность. Эти «обратные» или «связующие» сигнатуры встроены в подписи самопроверки, которые GnuPG добавляет к разделам подписи — просмотреть их с помощью `--list-sigs` нельзя.

В более старых версиях GnuPG или других приложениях OpenPGP эта функция может отсутствовать,

а подписывающие подключа, созданные этими приложениями, могут не иметь необходимых сигнатур привязки. Владельцы таких ключей могут решить эту проблему с помощью команды `cross-certify`, доступной в режиме редактирования ключей последующего `gpg`. Официальное пояснение см. на www.gnupg.org/faq/subkey-cross-certify.html.

Одной пары подключей достаточно для повседневного использования на всех ваших устройствах, но вы можете создать их для каждого в отдельности, если хотите защититься от одиночной потери. Повторите те же шаги, чтобы создать столько угодно дополнительных подключей для подписей, однако ключ шифрования на всех устройствах лучше использовать один и тот же, иначе тот, кто захочет отправить зашифрованный материал, не будет знать, какой ключ выбрать (ведь он отправляет его вам, а не вашему телефону, планшету или ноутбуку).

Если вы уже создали подключа для отдельных устройств, можете использовать временные связки, распределив ключи по устройствам. Вы импортируете свои подключа и удаляете ненужные (которые будут оставаться в сохранности в импортированном файле), а затем экспортируете, как и прежде, но в новый файл. Помните, что каждому устройству необходимы два секретных ключа: общий ключ шифрования и ключ подписи устройства. Используйте режим редактирования для выбора и удаления остальных подключей.

```
$ gpg --homedir .gnupg-temp --import subkeys.gpg
$ gpg --homedir .gnupg-temp --edit-key alice
gpg> key 3
gpg> delkey
gpg> save
$ gpg --homedir .gnupg-temp --export-secret-subkeys > device1-keys.gpg
```

Перезагрузите подключа и повторите процесс для каждого устройства, для которого хотите экспортировать ключи.

В целях безопасности, основной ключ стоит хранить офлайн — в идеале, создав его на надежном компьютере, защищенном от сомнительного Интернета (и возможно, также от вашей чуть менее сомнительной локальной сети). Как вариант, можно воспользоваться Tails Linux и сохранить связку в его защищенном постоянном томе. Дополнительную информацию об этом см. на сайте https://tails.boum.org/doc/first_steps/persistence.

Еще одно хорошее решение, которого, пожалуй, будет достаточно для не самых конченых параноиков, это удалить первичный ключ из связки, когда он не требуется, и сохранить где-нибудь его защищенную резервную копию. Для этого нужно удалить ключ полностью, а затем импортировать только подключа.

Прежде чем удалить первичный ключ, вы должны его экспортировать, помня, что экспортированный файл будет единственной копией вашего основного секретного ключа до тех пор, пока вы не сделаете резервную копию.

```
$ gpg --export-secret-keys alice > primary.gpg
$ gpg --delete-secret-key alice
$ gpg --import subkeys.gpg
```

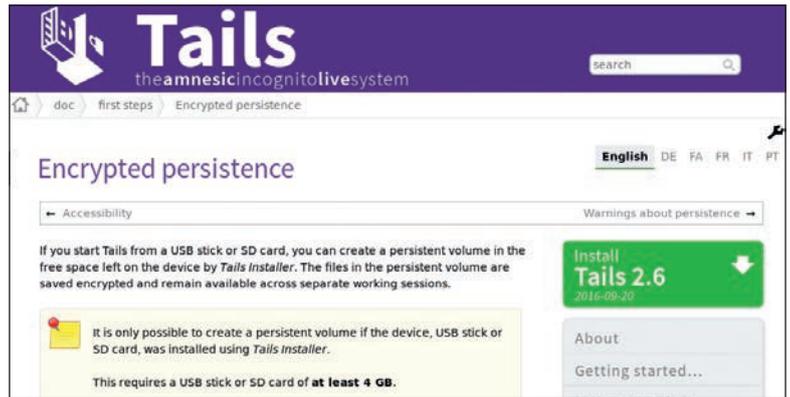
Не забывайте, что если вы удалите первичный ключ, то для его использования вам надо будет его импортировать (а затем снова удалить):

```
$ gpg --import primary.gpg
```

Отозвать и восстановить

Ваша связка ключей — это ваше сокровище. Вам надо ее защищать, но вы должны быть готовы и уничтожить ее, если случится худшее. Вам нужна безопасная резервная копия, а также сертификат отзыва. Это специальный сертификат, который применяется, чтобы оповестить всех, что вашему ключу нельзя доверять и, следовательно, его нельзя использовать. Как только это произойдет, сообщения не смогут быть подписаны или зашифрованы. Существующие сообщения можно будет расшифровать и подтвердить их достоверность, но GnuPG будет предупреждать пользователя о том, что ключ аннулирован.

GnuPG автоматически создает сертификат отзыва вместе с созданием ключа; найти его можно будет в `~/gnupg/openpgp-revocs.d` с названием, совпадающим с отпечатком ключа плюс суффикс



`.rev`. Для защиты от случайного использования его необходимо предварительно изменить. Это текстовый файл, он содержит инструкции, объясняющие, что вам нужно сделать (удалить двоеточие в начале первой строки сертификата).

Вы можете самостоятельно создать сертификат отзыва, указав причину:

```
$ gpg --gen-revoke alice > alice.rev
```

Следуйте инструкциям, чтобы выбрать причину отзыва и добавить описание (по желанию). GnuPG выдает предупреждающее сообщение о том, что сертификат отзыва надо хранить в безопасности — ведь они столь же ценны, как и закрытый ключ, который они могут сделать недействительным.

Если вам когда-либо понадобится использовать сертификат отзыва, просто импортируйте его в связку ключей:

```
$ gpg --import alice.rev
```

Вы также можете отозвать ключ напрямую, используя команду `revkey` в режиме редактирования. Таким образом можно аннулировать один или нескольких ключей, а также отозвать подключа без отзыва первичного ключа. Существуют также опции `revuid` и `revsig`, которые служат для отмены идентификаторов пользователей и подписей.

После отзыва, равно как и внесения любого другого изменения в связку ключей, которую вы хотите опубликовать, загрузите связку на сервер:

```
$ gpg --send-key alice
```

Существуют и другие способы защиты вашего закрытого ключа и его сертификата отзыва: например, сохранить их печатные копии в безопасном месте, разбить их на части и сохранить в нескольких местах или предоставить нескольким доверенным лицам. *Paperkey* (www.jabberwocky.com/software/paperkey) — это инструмент, который извлекает секретные части вашего ключа (см. рис. справа на стр. 67) и печатает их с данными контрольной суммы, что при необходимости упрощает ввод вручную:

```
$ paperkey --secret-key primary.gpg --output private.txt
```

Восстановить их ничуть не сложнее — для этого используется ваш открытый ключ:

```
$ paperkey --pubring public.gpg --secrets private.txt --output secret.gpg
```

Восстановленный ключ по-прежнему защищен вашей кодовой фразой, чтобы кто-то из доверенных лиц смог воспользоваться им на случай печального исхода. *Ssss* или *gfsplit* реализуют “Shamir’s Secret Sharing” (да, это тот самый Shamir, известный по RSA [аббревиатура от фамилий Rivest, Shamir и Adleman — криптографический алгоритм с открытым ключом, — прим. пер.]), который позволяет разбить что-нибудь на несколько частей так, чтобы для восстановления оригинала достаточно было иметь, скажем, три части из пяти. Можно разделить свой ключ, сертификат отзыва, кодовую фразу и т.д., и передать их по отдельности тем, кому вы доверяете, чтобы они могли бы воспользоваться этими данными без вас. 

► Чтобы ваша связка ключей была надежной, можно использовать Tails Linux.

Скорая помощь

Tails Linux (tails.boum.org) является идеальным средством для управления ключами в автономном режиме, если хранить вашу связку ключей в постоянном [persistent] томе.

Скорая помощь

Вы можете ссылаться на ключ, используя идентификатор пользователя, короткий или длинный идентификатор ключа или его отпечаток. Для некоторых инструментов может потребоваться префикс “Ox”.

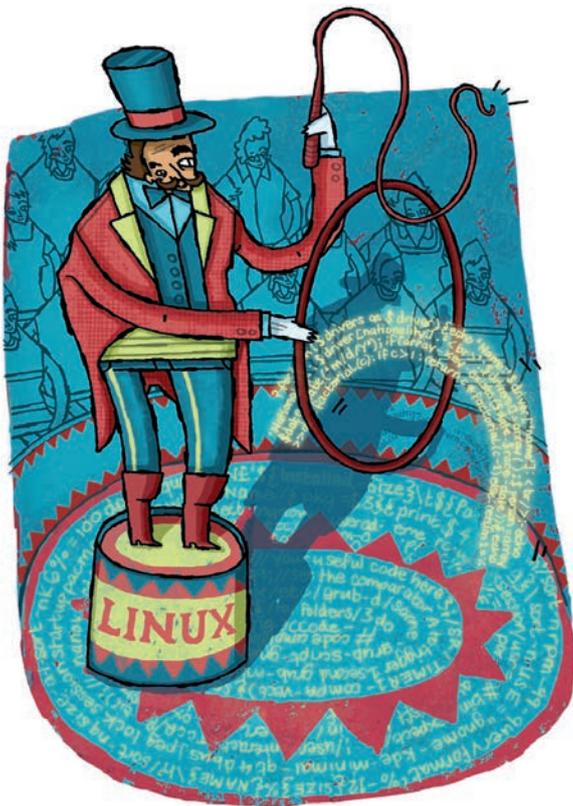
Varnish: Обратный прокси-сервер

Михалис Цукалос научит вас установить и настроить кэш Varnish HTTP, чтобы ускорить ваш сайт.



Наш эксперт

Михалис Цукалос — администратор Unix и баз данных, программист и математик, наслаждающийся написанием статей и узнаванием чего-то нового.



Скорая помощь

Возможно, вы задаетесь вопросом, стоит ли вам использовать Varnish, если вам не нужно обслуживать большое количество сайтов. Хотя, возможно, сильно улучшить вы не увидите, это очень полезно в целях обучения — установить и узнать, как его использовать, так что вперед — попробуйте, даже если вы обслуживаете всего лишь один сайт.

Varnish — это кэширующий обратный HTTP прокси-сервер. Он работает как прозрачный слой между вашим web-сервером и его web-клиентами, посредством кэширования контента, чтобы доставить его быстрее, не заставляя web-клиент ждать web-сервер. Другими словами, Varnish не может работать сам по себе, он должен находиться перед правильно настроенным web-сервером. Наиболее значимым преимуществом Varnish является то, что он уменьшает время отклика ваших сайтов.

Наш урок продемонстрирует основную функциональность Varnish; однако Varnish умеет делать намного больше, чем мы здесь показываем. И хотя он может работать на *Nginx*, в данном руководстве мы будем использовать сервер *Apache* — использование *Nginx* [см. Учебники LXF222] вместо *Apache* не является сложным,

Varnish HTTP Cache

I'm new here, please explain this Varnish thing

Questions and Answers

What is happening

2017-01-05 - Source Code Coverage

I usually brag that we have 90%+ automatic test coverage in Varnish, and that comes with a couple of footnotes.

The main one is that we don't automatically test the cursed(3) based tools (varnish[hist|stat|...] - only because I haven't found a way to do so.

For 2017 I've raised the bar: from now our goal is 90%+ on all our source code.

At the same time, I have retooled our code coverage test scripts so the *GCov* results can be seen online.

Presently at 81.8%, but we'll get there...

/phk

2016-12-21 - Project server upgraded

Now running FreeBSD 11-RELEASE /phk

2016-12-02 - Maintenance release of 4.1.4, 4.0 is End Of Life

Varnish Cache 4.1.4 has been released and is ready for download. This is a maintenance release with bug fixes and performance enhancements.

In other news, Varnish Cache 4.0 has reached End Of Life, and with this we have released Varnish Cache 4.0.4. This contains all the last fixes from git, and there will be no 4.0.5. Please up-

Index

News

Download

VMODS & Extras

Get help

FAQ

Mailing Lists

Security

Documentation

trunk | 5.0 | 4.1 | 4.0

Varnish Book (PDF)

4.2 | 3.0

GitHub project

Tips and tricks

Varnish Business

VTEST | GCov

Hosted by:

RootBSD

➤ Это сайт Varnish, где можно найти последние новости и документацию Varnish.

если вы знаете, как изменять настройку *Nginx*. Вы можете раздобыть дополнительную информацию о Varnish, посетив официальный сайт на <https://varnish-cache.org>. Часть этого сайта показана на скриншоте вверху этой страницы.

Важное предупреждение: настоятельно рекомендуем экспериментировать с Varnish на тестовой или виртуальной машине, поскольку неправильно настроенный процесс Varnish может ограничить доступность ваших сайтов для остального мира.

Varnish вообще

Установить Varnish в Debian не сложнее, чем выполнить следующую команду с правами root:

```
# apt-get install varnish
```

Вы можете узнать используемую вами версию Varnish, выполнив такую команду:

```
# varnishd -V
```

```
varnishd (varnish-4.0.2 revision bfe7cd1)
```

```
Copyright (c) 2006 Verdens Gang AS
```

```
Copyright (c) 2006-2014 Varnish Software AS
```

Установка Varnish включает множество исполняемых файлов: **varnishadm**, **varnishhist**, **varnishncsa**, **varnishtest**, **varnishd**, **varnishlog**, **varnishstat** и **varnishstop**.

О кэшировании HTTP

HTTP-кэширование — это способ хранения web-трафика, чтобы люди, подключающиеся к сайту, использовали кэширующий сервер вместо сайта, тогда всё будет быстрее. Причины, по которым мы используем HTTP-кэширование, во-первых, в том, что клиентам будет казаться, что их интернет-соединение работает быстрее, и во-вторых, чтобы вы могли преобразовывать и фильтровать ваш трафик. Побочным эффектом HTTP-кэширования является то, что клиенты всё еще могут загрузить web-страницу, даже если сайт на самом деле не работает — при условии, что страница успела сохраниться в кэше.

Прокси-сервер взаимодействует с клиентами и оконечным web-сервером следующим образом. Когда прокси-сервер получает запрос, который можно обслужить с помощью его кэша памяти или кэша жесткого диска, сообщение запроса не идет дальше к конечному получателю, а прокси-сервер возвращает запрашиваемые файлы, хранящиеся в его кэше. Похожим образом, когда прокси-сервер получает запрос, который согласно одному из его списков управления доступом должен быть отклонен, сервер не обслуживает запрос и блокирует доступ.

Если вы «счастливы», и желаемые данные находятся в кэше, то это называется попаданием [hit], в ином случае — промахом [miss]. «Промашнувшись», вы можете получить желаемые данные, возможно, чуть попозже, поскольку поиск по кэшу означает дополнительную задержку. Также есть пропуски [passes] — когда запросы игнорируются и никогда не кэшируются, и каналы [pipes], которые стабилизируют прямое подключение к движку и в основном используются для потокового мультимедиа. Пожалуйста, учтите, что пропуски не влияют на частоту попаданий.

Основная директория конфигурации Varnish — это `/etc/varnish`, но относящиеся к Varnish файлы также есть в `/etc/default`, как можно увидеть:

```
$ ls -l /etc/default/varnish*
-rw-r--r-- 1 root root 3768 Oct 14 2014 /etc/default/varnish
-rw-r--r-- 1 root root 514 Oct 14 2014 /etc/default/varnishlog
-rw-r--r-- 1 root root 799 Oct 14 2014 /etc/default/varnishncsa
```

Среди этих файлов самыми важными файлами конфигурации являются `/etc/varnish/default.vcl` и `/etc/default/varnish`, поскольку именно в этих двух местах определяются наиболее важные параметры Varnish.

Для работающего процесса Varnish единственный важный параметр — это размер его кэша, поскольку он влияет на производительность. Однако на нашем уроке мы будем придерживаться размера кэша по умолчанию.

Varnish по-простому

Varnish запускается следующим образом:

```
# service varnish start
# service varnishlog start
# service varnishncsa start
```

Аналогично вы можете прекратить все процессы Varnish:

```
# service varnish stop
# service varnishlog stop
# service varnishncsa stop
```

Следовательно, выполнив `ps ax | grep varnish` после запуска Varnish, вы увидите как минимум три явных работающих процесса: процесс `varnishd`, выполняющий всю работу по поддержке функциональности Varnish, сервис `varnishlog`, выполняющий утилиту `varnishlog` в качестве сервиса, чтобы позаботиться о файлах логов Varnish, и сервис `varnishncsa`, похожий на сервис `varnishlog`, но записывающий его файлы логов в формате NCSA Common Log Format.

Вы легко можете определить способ, которым сервис Varnish (`varnishd`) был выполнен:

```
$ ps ax | grep varnishd | head -1
825 ? Ss 0:00 /usr/sbin/varnishd -j unix,user=vcache -F -a :6081 -T localhost:6082 -fetc/varnish/default.vcl -S /etc/varnish/secret -s malloc,256m
```

Итак, предыдущий экземпляр Varnish использует порт номер 6081 (`-a:6081`), в основном применяемый для тестов. Чтобы Varnish взял на себя управление трафиком вашего сайта, ему следует слушать TCP-порт, которым раньше владел web-сервер, а web-сервер должен выбрать другой доступный порт. Другими словами, Varnish будет управлять всеми HTTP-запросами и обрабатывать их.

В результате настало время изменить текущую настройку на вашем сервере Linux и заставить Varnish слушать TCP-порт 80, а web-сервер Apache — слушать TCP-порт 8080, выбранный потому, что его легко запомнить. В системе Debian Linux вам вначале нужно внести изменения в два файла конфигурации Varnish, как показано в следующем выводе утилиты `diff`:

```
# cd /etc/default/
# diff varnish varnish.orig
48c48
< DAEMON_OPTS="-a :80 \
---
> DAEMON_OPTS="-a :6081 \
# cd /etc/varnish/
# diff default.vcl default.vcl.orig
18c18
< .port = "8080";
---
> .port = "8888";
```

Первое изменение велит Varnish использовать порт номер 80, тогда как второе сообщает Varnish, что его клиент — в данном

Скорая помощь

Вы можете узнать больше о Varnish, прочитав о нем бесплатную книгу, которую можно найти на <https://info.varnish-software.com/the-varnish-book>, и техническую документацию на <https://info.varnish-software.com/white-papers>.

```
2. mtsouk@mail: ~ (ssh)
root@mail:/etc/default# tail -5 /var/log/varnish/varnishncsa.log
2.86.25.121 - - [17/Jan/2017:19:40:42 +0200] "GET http://www.mtsoukalos.eu/sites/default/files/LUDI69.png HTTP/1.1" 200 337683 "http://www.mtsoukalos.eu/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.95 Safari/537.36"
2.86.25.121 - - [17/Jan/2017:19:40:45 +0200] "GET http://www.mtsoukalos.eu/sites/all/themes/corporateclean/images/inactive-slide.png HTTP/1.1" 200 3197 "http://www.mtsoukalos.eu/sites/all/themes/corporateclean/style.css?oijs62" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.95 Safari/537.36"
2.86.25.121 - - [17/Jan/2017:19:40:45 +0200] "GET http://www.mtsoukalos.eu/sites/all/themes/corporateclean/images/active-slide.png HTTP/1.1" 200 674 "http://www.mtsoukalos.eu/sites/all/themes/corporateclean/style.css?oijs62" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.95 Safari/537.36"
2.86.25.121 - - [17/Jan/2017:19:40:45 +0200] "GET http://www.mtsoukalos.eu/misc/favicon.ico HTTP/1.1" 200 5430 "http://www.mtsoukalos.eu/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.95 Safari/537.36"
180.76.15.150 - - [17/Jan/2017:19:41:01 +0200] "GET http://mtsoukalos.eu/node/156 HTTP/1.1" 200 6072 "-" "Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)"
root@mail:/etc/default# tail -15 /var/log/varnish/varnish.log
- Debug "RES_MODE 2"
- RespHeader Connection: close
- RespHeader Accept-Ranges: bytes
- Timestamp Resp: 1484674861.580149 0.089336 0.000070
- Debug "XXX REF 2"
- ReqAcct 223 0 223 665 6072 6737
- End

* << Session >> 131121
- Begin sess 0 HTTP/1
- SessOpen 180.76.15.150 13192 :80 109.74.193.253 80 1484674861.490754 15
```

Здесь вы можете видеть формат данных, хранящихся в `varnish.log` и в `varnishncsa.log`, а также небольшую часть вывода утилиты `varnishlog`.

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

Скорая помощь

Если вы захотите узнать больше о web-протоколах и о работе HTTP, здесь есть много информации. Мы рекомендуем *Высокопроизводительные браузерные сети* [High Performance Browser Networking] от Ильи Григорика [Ilya Grigorik] и *Web-протоколы и Практика: HTTP/1.1, Сетевые Протоколы, Кэширование и Измерение Трафика* [Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement] от Балачандера Кришнамурти [Balachander Krishnamurthy] и Дженнифер Рексфорд [Jennifer Rexford].

случае, *Apache* — слушает TCP-порт 8080. Вдобавок вам понадобится велеть Apache Virtual Sites, слушаемому порт номер 80, отныне слушать TCP-порт 8080. Это также потребует изменения значения параметра Listen в файле `/etc/apache2/ports.conf`. Если у вас есть другие виртуальные хосты *Apache*, не использующие TCP-порт 80, в их конфигурацию не надо вносить никаких изменений!

И напоследок, вам еще надо выполнить следующие действия:

```
# cp /lib/systemd/system/varnish.service /etc/systemd/system
# vi /etc/systemd/system/varnish.service
# diff /lib/systemd/system/varnish.service /etc/systemd/system/varnish.service
9c9
< ExecStart=/usr/sbin/varnishd -a :6081 -T localhost:6082 -f /etc/varnish/default.vcl -S /etc/varnish/secret -s malloc,256m
---
> ExecStart=/usr/sbin/varnishd -a :80 -T localhost:6082 -f /etc/varnish/default.vcl -S /etc/varnish/secret -s malloc,256m
# systemctl daemon-reload
# systemctl restart apache2.service
# systemctl restart varnish.service
```

Последние две команды используются для перезапуска двух сервисов. Вначале вам следует перезапустить *Apache*, чтобы освободить TCP-порт 80, а затем Varnish, который займет TCP-порт 80. Итак, следующий вопрос: «Как я могу быть уверен, что я использую Varnish?» Ответ простой: попытайтесь получить доступ к одному из сайтов *Apache* — и увидите, работает ли он!

Вдобавок вы можете проверить лог-файлы Varnish, одну из различных его утилит или вывод следующей команды:

```
# ps ax | grep varnishd | grep -v grep | tail -1
7883 ? Sl 0:01 /usr/sbin/varnishd -a :80 -T localhost:6082 -f /etc/varnish/default.vcl -S /etc/varnish/secret -s malloc,256m
```

Если значение параметра `-a:80`, то Varnish настроен правильно.

Пожалуйста, помните, что выдача Varnish слишком много места на диске или ОЗУ, возможно, не даст прироста производительности,

```
list length 308
8.09 VCL_return fetch
8.09 VCL_return deliver
5.71 RespHeader Accept-Ranges: bytes
4.17 RespStatus 200
4.17 BerespStatus 200
4.17 ObjStatus 200
4.17 RespReason OK
4.17 BerespReason OK
4.17 ObjReason OK
4.17 ReqMethod GET
4.17 BereqMethod GET
4.17 VCL_call HASH
4.17 VCL_call RECV
4.17 VCL_call PASS
4.17 RespHeader Age: 0
4.17 VCL_return pass
4.17 ReqProtocol HTTP/1.1
4.17 RespProtocol HTTP/1.1
4.17 BereqProtocol HTTP/1.1
4.17 BerespProtocol HTTP/1.1
4.17 ObjProtocol HTTP/1.1
4.17 VCL_call DELIVER
4.17 Debug XXX REF 1
4.17 VCL_return lookup
```

➤ Скриншот, показывающий утилиту *varnishstop* в действии. Вы можете изменить его вывод, используя соответствующие опции командной строки.

поскольку это замедлит поиск по кэшу. В зависимости от ваших данных, около 4 ГБ места на диске может стать хорошей отправной точкой. Поиск оптимального размера кэша — это искусство, требующее постоянного отслеживания производительности Varnish с помощью таких инструментов, как *varnishstat* и *varnishstop*. И напоследок, вам следует понимать, что весь процесс прозрачен для web-клиентов, что крайне важно.

Также помните, что очень удобно иметь резервную копию каждого файла настройки, который вы изменяете: ведь иногда что-то способно пойти не так. В данном руководстве у исходных файлов расширение `.orig`.

Журналы Varnish

В системе Debian лог-файлы Varnish можно найти в папке `/var/log/varnish/`. На самом деле, есть два лог-файла, с именами **varnish.log** и **varnishncsa.log**. Файл **varnish.log** хранит данные логов Varnish в формате, определенном Varnish, что более читаемо, но занимает больше экранного места, тогда как **varnishncsa.log** использует знакомый формат журналов, используемый в лог-файлах *Apache*.

Скриншот на стр. 71 демонстрирует данные в **varnish.log** и **varnishncsa.log**, чтобы дать вам лучшее понимание того, какие типы данных в них хранятся. Утилита *varnishlog* позволяет увидеть лог-файлы Varnish в лучшем формате.

Очень полезно знать, что *Apache* всё еще записывает предполагаемые лог-файлы в свои собственные лог-файлы, так что у вас не будет никаких потерь трафика.

Утилита *varnishncsa* отображает данные журнала, записанные в **varnishncsa.log**, а утилита *varnishstat* предназначена для тестирования кэша Varnish. Вы можете найти больше информации о них, взглянул на их map-страницу.

Скриншот слева внизу — вывод map-страницы *varnishstop*, работающей как утилита командной строки верхнего уровня, но для файлов Varnish. Однако рисунок сверху куда интереснее, поскольку показывает *varnishstop* в реальном действии.

Инструмент *varnishadm* позволяет управлять запущенным процессом Varnish, а инструмент *varnishhist* показывает гистограмму

```
VARNISHTOP(1)                                VARNISHTOP(1)
NAME
  varnishstop - Varnish log entry ranking

SYNOPSIS
  varnishstop [-1] [-b] [-c] [-C] [-d] [-f] [-g <session|request|vxd|raw>] [-h] [-i taglist] [-I <[taglist:]regex>] [-L limit] [-n name] [-N filename] [-p period] [-q query] [-r filename] [-T seconds] [-x taglist] [-X <[taglist:]regex>] [-V]

DESCRIPTION
  The varnishstop utility reads varnishd(1) shared memory logs and presents a continuously updated list of the most commonly occurring log entries. With suitable filtering using the -I, -i, -X and -x options, it can be used to display a ranking of requested documents, clients, user agents, or any other information which is recorded in the log.

  The following options are available:

  -1      Instead of a continuously updated display, print the statistics once and exit. Implies -d.
  -b      Only display transactions and log records coming from backend communication.
  -c      Only display transactions and log records coming from client communication.
  -C      Do all regular expression and string matching caseless.
  -d      Start processing log records at the head of the log instead of the tail.
  -f      Sort and group only on the first field of each log entry. This is useful when displaying e.g. stataddr entries, where the first field is the client IP address.
```

➤ Map-страница утилиты командной строки *varnishstop* показывает постоянно обновляемый список наиболее частых записей в журнале Varnish.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

запросов Varnish, весьма полезную для оценки производительности Varnish.

Как вы уже знаете, Varnish поставляется со многими полезными утилитами, информирующими о его работе и производительности. Поговорим об утилите *varnishstat*, отображающей статистику запуска процесса Varnish, используя различные показатели, предоставляемые Varnish.

Varnishstat может конвертировать свой вывод в XML, при параметре `-x`, или в формат JSON, при параметре `-j` — вы можете увидеть оба формата вывода на рис. внизу. Поскольку структурированный вывод не особенно удобен для проверки в реальном времени, полезность вывода XML и JSON заключается в том, что он позволяет хранить сгенерированные данные в базе данных, чтобы запросить их позднее. Можно также получить вывод в текстовом формате, через опцию `-l`. Если вы не укажете ни одну из этих трех опций, *varnishstat* будет работать так же, как утилита *top*, и автоматически обновлять свой вывод, пока вы не завершите ее работу, нажав клавишу `Q`.

Секреты Varnish

HTTP/2 — это последняя версия протокола HTTP, сконцентрированная на производительности. Ключевые отличия HTTP/2 от HTTP/1.1 — это передача данных как бинарник вместо текста, безопасность благодаря TLS, многоканальная передача, позволяющая HTTP/2 использовать одиночное TCP-подключение для распараллеливания, сжатие заголовков и поддержка 'push'-ответов. Большинство из вышеупомянутых различий и улучшений делают HTTP/2 значительно быстрее, чем HTTP/1.1, но все прежние методы HTTP, идентификаторы состояния и семантика в HTTP/2 остаются теми же. Возможно, вы спросите, как же отлаживать подключения HTTP/2, если они зашифрованы. Решение в том, чтобы использовать NSS-кейлоггинг вместе с плагином Wireshark — на данный момент это работает с web-браузерами *Chrome* и *Firefox*. Вы можете узнать больше об HTTP/2 на <http://httpwg.org/specs/rfc7540.html>.

TLS означает Transport Layer Security [безопасность транспортного уровня], и ее главная цель — обеспечение безопасности и приватности взаимодействия двух приложений. Вся подноготная TLS-протокола показана на его RFC, который находится на <https://tools.ietf.org/html/rfc5246>.

Чтобы программа вроде Varnish осуществляла кэширование, нужны две основные вещи: способ решить, надо ли кэшировать ресурс, и эффективный способ получить доступ к кэшу и его эксплуатировать. Другими словами, нет смысла кэшировать динамическую страницу, которая каждую секунду меняется, и нет смысла в наличии данных кэша, слишком медленных для поиска или требующих слишком много времени для сохранения.

Varnish будущего

На нашем уроке мы используем Varnish версии 4, но уже доступен Varnish 5, предлагающий кое-какие новые важные функции. Самая важная функция Varnish, находящаяся в данный момент

в экспериментальном состоянии — это поддержка HTTP/2, которая позволяет протестировать трафик HTTP/2 и увидеть, как он ладит с Varnish. Но помните, что по умолчанию поддержка HTTP/2 в Varnish 5 отключена. Другое новое свойство Varnish 5 — переход с графика выпуска функций на график выпуска релизов, т.е. вы можете ожидать новую версию Varnish каждые шесть месяцев. Поскольку Varnish 5 вышел в сентябре 2016 г., следующая его версия — от марта 2017 г. Varnish 5 также поддерживает TLS-ограничители, прерывающие TLS-подключения для вас, и протокол PROXY. Если вы захотите получить поддержку двух предыдущих функций, вы можете использовать Hitch TLS (<https://hitch-tls.org>).

Varnish 5 также включает новый контроллер под названием Shard, ответственный за распределение нагрузки, с поддержкой Negative Cache, это очень продвинутая функция. В Varnish 5 еще много новых функций, но разговоры о них выходят за рамки нашего урока.

Помните, что вам не обязательно задействовать все возможности Varnish, в основном потому, что некоторые из них узко специализированы. Общая идея в том, чтобы использовать то, что вам действительно нужно, прежде чем перейти к более продвинутым функциям, чтобы действительно понимать, что вы делаете. **LXF**

```

2. mtsouk@LTTng: ~ (ssh)
mtsouk@LTTng:~$ sudo varnishstat -x | head -18
<?xml version="1.0"?>
<varnishstat timestamp="2017-01-16T10:25:34">
  <stat>
    <type>MAIN</type>
    <name>uptime</name>
    <value>130</value>
    <flag>c</flag>
    <format>d</format>
    <description>Child process uptime</description>
  </stat>
  <stat>
    <type>MAIN</type>
    <name>sess_conn</name>
    <value>0</value>
    <flag>c</flag>
    <format>i</format>
    <description>Sessions accepted</description>
  </stat>
mtsouk@LTTng:~$ sudo varnishstat -j | head -18
{
  "timestamp": "2017-01-16T10:25:38",
  "MAIN.uptime": {
    "description": "Child process uptime",
    "type": "MAIN", "flag": "c", "format": "d",
    "value": 134
  },
  "MAIN.sess_conn": {
    "description": "Sessions accepted",
    "type": "MAIN", "flag": "c", "format": "i",
    "value": 0
  },
  "MAIN.sess_drop": {
    "description": "Sessions dropped",
    "type": "MAIN", "flag": "c", "format": "i",
    "value": 0
  },
  "MAIN.sess_fail": {
mtsouk@LTTng:~$ which varnishstat

```

► Утилита командной строки *varnishstat*, которая отображает подробный снимок текущей производительности Varnish и умеет генерировать вывод в XML и JSON.

Varnish против Squid

Squid — это очень популярный прокси-сервер переадресации, который можно настроить, чтобы он работал как обратный прокси-сервер. Однако Varnish разрабатывался как обратный прокси-сервер изначально, с самого начала. Вдобавок Varnish имеет лучшую систему настройки, позволяющую переписывать URL, плюс лучшую производительность и масштабирование, поскольку Varnish является

многопоточным приложением, а Squid использует только один процесс. И последнее преимущество — у Varnish предусмотрена лучшая поддержка аннулирования.

Зато у Squid имеется встроенная поддержка SSL, тогда как Varnish требует для этого дополнительной настройки. Также Squid поддерживает антивирусные плагины, что для определенных пользователей

может быть действительно важно. И наконец, Squid более зрелый, чем Varnish, поскольку впервые появился в июле 1996 г.

Если упомянутые различия не играют для вас большой роли, значит, вы можете работать в любом из этих приложений!

Подробности о Squid вы найдете на www.squid-cache.org.

ЧАСТЬ 6

IoT: Дом, который послушен нам

Максим Черепанов продолжает воспитывать наше жилище, превращая его в Послушный Дом.



Наш эксперт

Максим Черепанов — заместитель директора в небольшой коммерческой фирме. По образованию железнодорожник-управленец. Linux занимается с 2008 г. Чем дольше это делает, тем дальше от Windows. За эти годы разучился чистить реестр и забыл слово «антивирус».



Мы расстались на том волнующем моменте, когда командный контроллер сумел послать по радиоканалу на периферийный контроллер (Arduino) контрольный пакет (пинг) и получил назад тот же пакет. Покопавшись в коде, вы можете обнаружить, что это происходит в функциях `radio.read()` и `radio.write()`. Замечу, что функции интересные: они записывают значения не в переменные, а непосредственно в области памяти по указателям (&).

Но сейчас мы занимаемся не курсом C/C++, а практикой. И потому сразу перейдем к делу. Я не сторонник академического изучения программирования — по какой-то причине поспешил сразу извлечь из имеющихся заготовок (`pingtest.cpp` и `scanner.pde`) экстракт, необходимый для нас и нашего ПД.

Программа для BananaPi

Если вы продолжаете следовать за мной, то создайте файл `nrf24.cpp` и вставьте в него следующее содержание (комментарии можно опустить, причем для экономии места я комментировал только важные моменты):

```
#include <cstdlib>
#include <iostream>
#include "../RF24.h"
```

```
RF24 radio("/dev/spidev0.0", 8000000, 25); // выбор нашего порта SPI, скорости и пина
const uint64_t pipes[2] = { 0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL };
// адреса "труб"
void setup(void) // настройка радиомодуля
{
    radio.begin();
    radio.setRetries(15,15); // сколько раз и через какой промежуток будут попытки достучаться до приемника
    radio.setAutoAck(true); // разрешим автоматическое подтверждение получения
    radio.setChannel(0x4c); // номер канала
    radio.setPALevel(RF24_PA_MAX); // максимальная мощность передатчика
    radio.openWritingPipe(pipes[0]); // "труба" для передачи
    radio.openReadingPipe(1,pipes[1]); // "труба" для приема
}
int main(int argc, char* argv[]) // точка входа в программу
{
    setup();
    radio.stopListening(); // останавливаем прослушку эфира
    unsigned long argument = atoi(argv[1]); // заносим в переменную argument числовое значение первого параметра, передаваемого в исполняемую программу
    radio.write( &argument, sizeof(unsigned long) ); // передаем argument в эфир
    radio.startListening(); // начинаем прослушку
    в ожидании ответа
    // в следующем фрагменте 1100 миллисекунд слушаем эфир в ожидании ответа
    // это подобрано экспериментально (разрядка конденсатора)
    // если ответа не будет, программа вернет "-1"
    unsigned long started_waiting_at = __millis();
    bool timeout = false;
    while ( ! radio.available() && ! timeout )
    {
        __msleep(5);
        if ( __millis() - started_waiting_at > 1100 ) timeout = true;
    }
    if ( timeout )
        return -1;
    // если ответ получен, то читаем его в переменную got_time
    else
    {
        unsigned long got_time;
        radio.read( &got_time, sizeof(unsigned long) );
    }
}
```

```
printf("Got response %lu\n\n",got_time);
}
return 0;
}
```

После записи компилируем проект — давайте уж без make, а ручками:

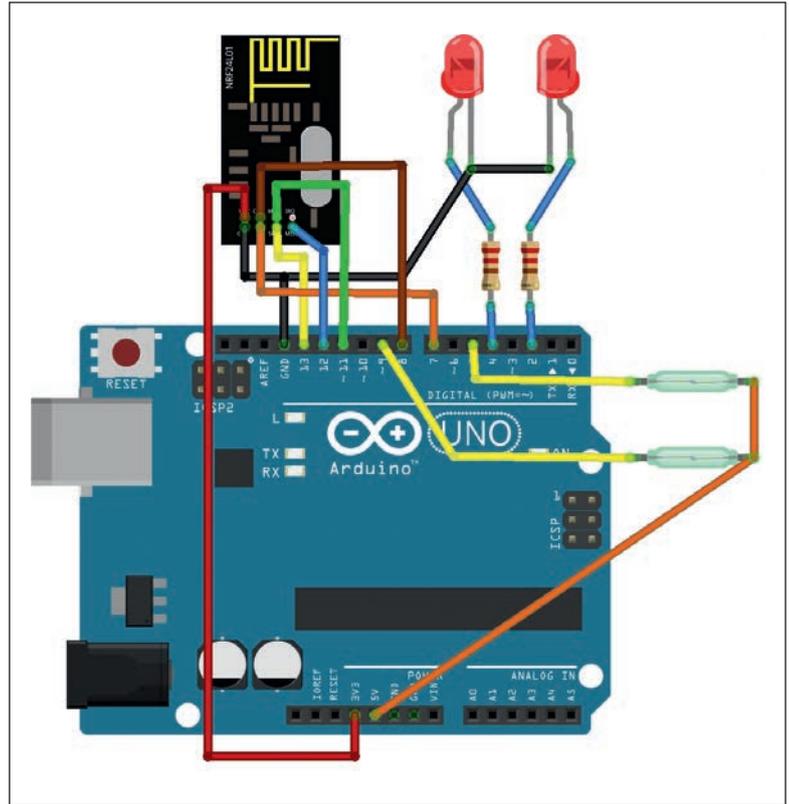
```
# g++ -L../librf24/ -g -lrf24 nrf24.cpp -o nrf24
```

Данную строку я честно позаимствовал из Makefile. При компиляции будет предупреждение о несоответствии типов; в данной ситуации это неважно.

Периферийный контроллер: Arduino

Я надеюсь, что с прошлой статьи у вас в библиотеках осталась RF24-master. Просто откройте IDE, очистите окно и вставьте данный скетч:

```
#include <RF24_config.h>
#include <RF24.h>
#include <nRF24L01.h>
#include <SPI.h>
// Установка пинов CE в 7 и CSN в 8
RF24 radio(7,8);
// Установка адресов труб
const uint64_t pipes[2] = {0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL};
void setup(void)
{ // Установка командных пинов для 2 каналов
pinMode(2, OUTPUT);
pinMode(4, OUTPUT);
// Установка контрольных пинов для 2 каналов
pinMode(5, INPUT);
pinMode(9, INPUT);
// Установка контроля на мониторе порта
Serial.begin(57600);
printf_begin();
// Настраиваем радиомодуль
radio.begin();
radio.setAutoAck(true);
radio.setRetries(15,15);
radio.setChannel(0x4c);
radio.setPALevel(RF24_PA_MAX);
radio.openWritingPipe(pipes[1]);
radio.openReadingPipe(1,pipes[0]);
radio.startListening();
radio.printDetails();
}
void loop(void)
{
if ( radio.available() )
{
// Ожидаем получения посылки
unsigned long command;
bool done = false;
while (!done)
{
// Получив, пишем в память
done = radio.read( &command, sizeof(unsigned long) );
// Печатаем в контрольный порт
printf("Got payload %lu...",command);
}
switch (command) {
case 30: digitalWrite( 2,LOW); break;
case 31: digitalWrite( 2,HIGH); break;
case 40: digitalWrite( 4,LOW); break;
```



► Рис. 1. Условная схема подключения.

```
case 41: digitalWrite( 4,HIGH);
}
delay(1000);
// Читаем состояние нагрузок
unsigned long control = digitalRead(5)*10 + digitalRead(9);
// Останавливаем слушание
radio.stopListening();
// Посылаем назад переменную control - состояние нагрузок
radio.write( &control, sizeof(unsigned long) );
printf("Sent response.\n\n");
// Стартуем заново слушание
radio.startListening();
}
}
int serial_putc( char c, FILE * ) {
Serial.write( c );
return c;
}
void printf_begin(void) {
fdevopen( &serial_putc, 0 );
}
```

Код прост, но всё же требует некоторых пояснений; взглянем на рис. 1. На этом рисунке я изобразил, как надо подключить к Arduino UNO радиомодуль (таблица подключений дана в прошлой части), и подключение нагрузок.

Чтобы вас не запутывать и не загромождать журнал, я нагрузки изобразил светодиодами (а они реально есть внутри опторазвязок), а контрольные фототранзисторы внутри оптопар — герконами (эти элементы имеются в наборе графического редактора *Fritzing*, о котором упоминалось в **LXF182** и др.; о нем мы поговорим в другой серии статей).

Как видите, нагрузки подключаются к цифровым выходам 2 (1-й канал) и 4 (2-й канал), а контрольные — к 9 (1-й канал) ►

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

и 5 (2-й канал), т.е. 2 и 4 — выходы, а 9 и 5 — входы. Именно это мы и конфигурируем в начале скетча. Остальное достаточно понятно и снабжено комментариями.

Теперь включите оба контроллера с подключенными радиомодулями, перейдите в директорию `BananaPi`, где у вас лежит скомпилированный файл `nrf24` (можете его перенести в удобное место), и выполните

```
# ./nrf24 31
```

Должен зажечься светодиод, который подключен к выводу 2. А если вы, как и я, подключили к Arduino исполнительное устройство на триаках, то у вас будет подано напряжение на 1-й канал нагрузки, и через секунду вы получите в терминале следующий ответ:

```
Got response 1
```

Теперь введите

```
#./nrf24 30
```

Напряжение с нагрузки снимется, и вы увидите

```
Got response 0
```

Если ввести `./nrf24 41` и `./nrf24 40`, то включится и отключится 2-й канал нагрузки, и вы получите в консоли ответами 10 и 0. А если включить одновременно обе нагрузки, в ответ придет 11.

Это уже настоящий контроль нагрузок: по ответу можно судить, какая из нагрузок включена.

Вы можете доработать код скетча так, чтобы при посылке другого значения (например, 99) состояние нагрузок не изменялось, а в ответ приходило их состояние.

Не удивляйтесь, что я применил именно такие числа для управления нагрузками — это сделано в расчете на учебную конфигурацию, ее обнаружим позже. А сейчас поговорим об одной проблеме, которая неожиданно испортила мне план написания очередной части, заставила пересмотреть алгоритм управления и познакомила с очередной интересной программой.

Проблемы прав

Ох уж эти права на файлы... Любой пользователь Windows, попробовав Linux чуть глубже, чем в качестве офисного ПК, с недоумением обнаруживает, что даже единственному пользователю тут можно не всё (если он не взял себе права администратора). И даже после многолетнего общения с Linux иногда случается получать сюрпризы от организации разграничения доступа к файлам и каталогам.

При написании файла управления для радиомодуля я заметил, что запуск возможен только от имени `root`. Вначале я не придал этому значения — мол, потом разберусь, пересоберу от имени обычного пользователя, и всё получится. Но, имея уже готовые заготовки, я никак не смог запустить управление NRF24 согласно плану. Сборка не от имени `root` ничего не дала: программа в процессе работы обращается к файлам, доступ к которым имеет только `root`. А мне необходимо было запустить управление из `web`-сервера, который работал от имени пользователя `www-data` (не имеющего `root`-привилегий).

Поиск решения по форумам не дал ничего: совершенно справедливо (как говорили многие форумчане), ни в коем случае нельзя запускать `Arache2` от суперпользователя, это недопустимая дыра в безопасности. Но в одном из сообщений я увидел спасительную для себя идею: при событии нажатия кнопки на `web`-странице в директории, доступной для записи (`/tmp`), должен создаваться файл, название или содержание которого должно быть признаком для процесса, управляющего NRF24, и после выполнения действия этот файл должен модифицироваться или стираться (файл-маркер). Звучит на первый взгляд сложновато, но дальше вы всё поймете.

Работа с файлами-маркерами

Для начала определимся, что следует запустить процесс-демон, работающий всё время, и раз в какой-либо промежуток времени (я принял раз в секунду) проверять наличие файлов-маркеров в `/tmp`. Написать постоянно работающую программу не сложно, для этого достаточно бесконечного цикла. Запуск ее при старте — дело *cron*tab, всё тоже прозаично. Но если процесс по какой-то причине прервется, то система по своей инициативе его не перезапустит... Что же делать?

На помощь приходит программа *Supervisor* — «демонизатор недемонизируемого». Это менеджер процессов, который существенно упрощает управление долго работающими программами, предоставляя простой и понятный интерфейс.

Ставится просто:

```
# apt-get install supervisor
```

Имеет в составе две утилиты, которые используются для организации процесса:

» *supervisord* — демон, который управляет (запуск/остановка/перезапуск) процессами;

» *supervisorctl* — скрипт, который общается с *supervisord* с помощью командной строки.

Давайте сначала напишем программу, которая будет работать постоянно, с правами `root`, и управлять модулем NRF24:

```
nano control.sh
#!/bin/bash
while true
do
test -e /tmp/11 && /home/bananapi/on.sh 0 && rm /tmp/11
test -e /tmp/10 && /home/bananapi/off.sh 0 && rm /tmp/10
test -e /tmp/21 && /home/bananapi/on.sh 1 && rm /tmp/21
test -e /tmp/20 && /home/bananapi/off.sh 1 && rm /tmp/20
test -e /tmp/31 && /home/bananapi/nrf24 31 && rm /tmp/31
test -e /tmp/30 && /home/bananapi/nrf24 30 && rm /tmp/30
test -e /tmp/41 && /home/bananapi/nrf24 41 && rm /tmp/41
test -e /tmp/40 && /home/bananapi/nrf24 40 && rm /tmp/40
sleep 1
done
```

Это бесконечный цикл, который запускается 1 раз в секунду (этого вполне достаточно: нам незачем перенапрягать систему). В первых четырех строках внутри цикла идет прямое управление пинами `BananaPi`, в последующих — нагрузками через модуль NRF24.

Допустим, в директории `/tmp` оказался файл-маркер с именем `31`, тогда сработает строка `test -e /tmp/31 && /home/bananapi/nrf24 31 && rm /tmp/31`. Что записано в ней? «Если файл-маркер `/tmp/31` существует, то запустить `nrf24` из домашней директории с аргументом `31`, и при успешном срабатывании удалить файл-маркер». Что значит — успешное? Если `nrf24` не вернет `-1` (см. листинг выше). Кстати, это упрощенная версия контрольного скрипта; к полной версии вернемся в последующих частях.

Я думаю, что скрипт можно написать короче и красивее. Но суть сейчас не в этом. Надо, чтобы скрипт работал постоянно, с правами `root`, и перезапускался при остановке. Внесем его в список файлов, управляемых *Supervisor*’ом.

Конфигурационные файлы *Supervisor* находятся в каталоге `/etc/supervisor/conf.d` (как правило, по файлу с расширением `.conf` на каждую программу). Простая конфигурация для нашего скрипта, сохраненная в `/etc/supervisor/conf.d/control.conf`, будет иметь такой вид:

```
[program:control]
command=/home/bananapi/control.sh
autostart=true
```

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

```
autorestart=true
stdout_logfile=/var/log/control.log
stderr_logfile=/var/log/control.log
user=root
numprocs=1
```

По порядку строк: имя программы, исполняемый файл, запуск при загрузке системы, перезапуск при падении, лог-файл срабатывания, лог-файл ошибок (последние две необязательны), пользователь, от которого запустится файл, сколько процессов может быть запущено одновременно.

Пробуем. Перезагружаем командный контроллер, проверяем запуск контрольного скрипта:

```
ps aux|grep control.sh
root 2073 0.1 0.1 2868 1236 ? S 20:58 0:06 /bin/bash /home/bananapi/control.sh
```

Как видим, скрипт запустился с нужными правами. Попробуем по новой?

```
touch /tmp/31
```

Создали файл-маркер, и тут же у меня загорелся контрольный светодиод, подключенный к выводу 2 Arduino, а в `/var/log/control.log` появилась строка

```
Got response 1
```

Это контрольное сообщение от радиомодуля: где-то на Arduino включилась нагрузка. Зайдите в директорию `/tmp`, и вы увидите, что файла с именем `31` уже нет.

Выключить можно так:

```
touch /tmp/30
```

Точно так же можно включить и отключить светодиод, подключенный к выводу 4, для этого надо создавать файлы `41` и `40` во временной директории.

Внесение изменений в web-сервер

Управлять нагрузкой, создавая файлы вручную, примитивно. Один или два раза это сделать можно, но постоянно — нудно и слишком неудобно. Заставим-ка делать эту черновую работу наш любимый *Apache*.

Давайте вспомним содержимое главного файла сервера — `/var/www/index.php` (фрагмент):

```
{
echo "<tr><th><form action='on.php' method='post'><p>
<button class='green' name='pin' value='\$pin'>Включить
\$A[\$pin]</button>
</form></th><th>";
include 'control.php';
echo "</th><th><form action='off.php' method='post'><p>
<button class='red' name='pin' value='\$pin'>Выключить
\$A[\$pin]</button>
</form></th></tr>";
\$pin++;
}
```

Как мы видим, данные о нажатой кнопке передаются во внешние файлы `on.php` (включение) и `off.php` (выключение). Давайте изменим их согласно новой идеологии управления нагрузками:

```
#nano /var/www/on.php
<html>
<head>
<title> Включение </title>
</head>
<body>
```

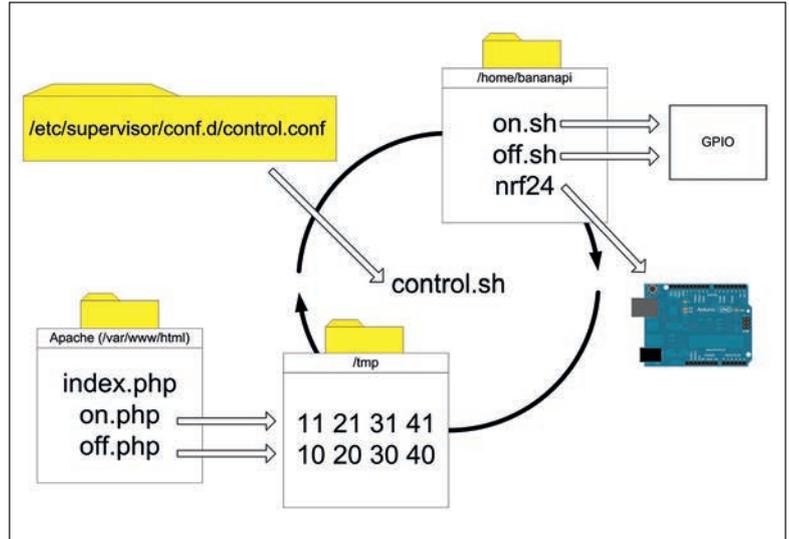


Рис. 2. Схема взаимодействия файлов и процессов.

```
<?php
\$pin=(\$ _POST['pin']+1)*10+1;
exec ("touch /tmp/\$pin");
header("Location: ".$ _SERVER["HTTP_REFERER"]);
?>
```

И фрагмент файла `off.php` (только часть, касающаяся создания файлов-маркеров):

```
\$pin=(\$ _POST['pin']+1)*10;
exec ("touch /tmp/\$pin");
```

Теперь в зависимости от нажатой кнопки в директории `/tmp` будут создаваться файлы для управления каналами нагрузок согласно таблице соответствия внизу страницы (десятки — номер канала, единицы — статус включения/выключения):

Отныне вам стали понятны имена файлов-маркеров? А обрабатывать их будет тот скрипт, что мы создавали ранее (`control.sh`).

Уф! Мы сегодня просто отлично поработали. Для закрепления достигнутого нарисуем общую схему взаимодействия файлов и процессов внутри нашего ПД на сегодняшний момент (рис. 2). Разумеется, можно сократить что-то: совместить файлы, убрать промежуточные звенья. Но вы обратили внимание, что для добавления нового канала управления достаточно будет просто перенаправить управляющие усилия в `control.sh`? Тогда можно гибко управлять конфигурацией, причем управление получается «читабельным».

Пару слов о контроле над скриптом `control.sh`. Он работает под управлением *Supervisor*'а, а самим супервизором можно «рулить» из системного интерфейса для управления и мониторинга *supervisorctl*. Выполнение команды — `supervisorctl <команда>`, самих команд у него много, упомяну наиболее востребованные:

- » `avail` Вывод всех доступных программ.
- » `start <name>` Введите `<name>`, чтобы запустить процесс или группу по имени, введите `all`, чтобы запустить все процессы.
- » `stop <name>` Аналогично предыдущему для остановки процессов.
- » `reread` Перечитать файл конфигураций, чтобы не перезапускать сам супервизор.
- » `status <name>` Вывести текущее состояние всех процессов либо конкретного процесса, передав его имя.

Этого вполне хватит, чтобы эффективно управлять демонизированными процессами. В следующей серии продолжим, а пока — пусть ваш дом растет послушным! LXF

	1-й канал	2-й канал	3-й канал	4-й канал
Включить	11	21	31	41
Выключить	10	20	30	40

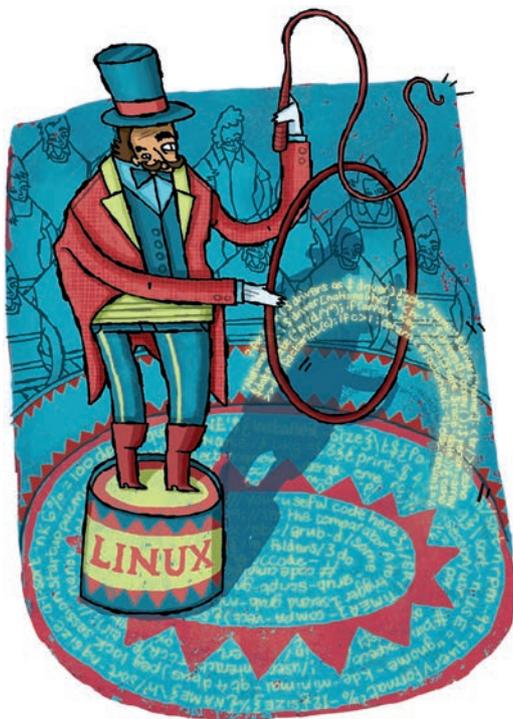
Cocos2D: Снова об играх

Михаил Вознесенский заинтересовался вековой борьбой добра со злом... то бишь, человека с компьютером.



Наш эксперт

Михаил Вознесенский — добропорядочный пользователь операционных систем семейства GNU/Linux, стремящийся показать, что работать с ними просто и свободно.



В выпуске журнала LXF210 углубленно рассказывалось о библиотеке *pygame*. Сегодня мы продвинемся дальше и перейдем от библиотеки к движку *cocos2d*. Принципиальная разница между ними заключается в том, что в движках многие алгоритмы уже заранее реализованы. Это позволяет сократить общее время разработки и сосредоточиться только на поставленной задаче.

Движок движку рознь. Можно выделить два вида движков. Одни распространяются с графической средой разработки, другие — без нее. Эти два факта часто являются ключевыми, особенно для новичков.

Общая структура построения приложения.



Движки, имеющие графическую среду разработки, пользуются большей популярностью, так как процесс разработки намного дружелюбнее к пользователю — настолько, что можно создать игру, не написав ни единой строчки кода. Что касается другой группы инструментария, то для ее применения необходимо изучать API (программный интерфейс приложений).

Прежде чем перейти к разработке, обратимся к основам, а именно — к системе координат и общей структуре построения приложения. Базовая схема архитектуры приложения представлена на рисунке. Она есть в документации к движку на официальной странице проекта.

Точкой отсчета системы координат является левый нижний угол окна приложения, а координаты расположения спрайта относятся к середине изображения, а не верхнему левому углу, как в *pygame*. Эта особенность позволяет нагляднее представить сцену.

Сценами являются части приложений (Вступление, Меню и т. д.) — их может быть много, но активна только одна. В этом заключается сильная сторона *cocos2d*. Получается, что при разработке программы основными частями, которыми приходится оперировать, являются сцены. В движке реализовано достаточно различных методов по их взаимодействию. В *pygame* такого нет, и чтобы писать многоуровневые приложения, для начала придется написать собственный модуль по организации работы, а уже потом приступить к написанию приложения.

Рассмотрим пару примеров для понимания работы примитивов в *cocos2d*.

Первая программа покажет, как создавать окно и выводить в нем текст:

```
import cocos
cocos.director.director.init(caption = "GameOver")
text = cocos.text.Label("Game Over !!!", position = (320, 240),
font_name = 'Golden Girdle', font_size = 20, color = (255, 0, 0, 255), anchor_x = 'center', anchor_y = 'center')
scene = cocos.scene.Scene()
scene.add(text)
cocos.director.director.run(scene)
```

Первая строка отвечает за инициализацию самого *cocos2d*. Ниже, создаем основное окно с заголовком "GameOver". По умолчанию, если явно не указать параметры, размер окна будет 640×480; также можно выставить параметр полноэкранного режима.

Далее создаем переменную *text*, содержимое которой будет выводиться на экран. За создание текста отвечает класс *Label* модуля *text* (не путать название переменной с модулем, это разные вещи). Этому классу были переданы следующие параметры: сам текст, заключенный в двойные (или одиночные) кавычки; координаты расположения текста в окне; имя используемого шрифта; размер шрифта; цвет текста (поддерживает альфа-канал);

и последние параметры — это так называемые якоря. Они определяют некую точку, относительно которой будет располагаться объект в окне (в текущем примере за такую точку взят центр текста; если эти якоря не задавать, то за точку, относительно которой будет идти вывод, принимается левый нижний угол текста).

После создания текста создается пустая сцена; ниже в нее добавляются объект `text` и затем выводят саму сцену в окно приложения.

Это был самый простой, базовый пример. Разберем пример посложнее. Здесь уже появятся спрайты плюс их передвижение и пересечение.

```
from cocos.director import director
from cocos.layer import Layer
from cocos.sprite import Sprite
from cocos.scene import Scene
import cocos.action as action
import cocos.collision_model as collision
```

Если обратить внимание на само написание кода в первом примере, то взаимодействие с движком происходит громоздко и неудобно. Для упрощения данного взаимодействия можно импортировать не весь движок, а только необходимую его часть. К примеру, в первых четырех строках содержится следующая формулировка: для соответствующего модуля импортировать соответствующий класс. После этого вместо `cocos.director.director` можно писать просто `director`.

Последние две строки совершают похожее действие, но слегка иначе. Для них формулировка выглядит так: импортируем соответствующий модуль как новое имя. Результат этих записей сводится к тому, чтобы вместо `cocos.action` употреблять просто `action`.

Отличие этих записей друг от друга заключается в том, что в первом варианте импортировался класс соответствующего модуля, а во втором варианте импортировался сам модуль, но с видоизмененным именем.

```
class ActorSprite(Sprite):
    def __init__(self, image, position, shift):
        super(ActorSprite, self).__init__(image, position)
        self.shift = shift
        motion = action.MoveBy((self.shift, 0), 5)
        self.do(action.Repeat(motion + action.Reverse(motion)))
        self.cshape = collision.AARectShape(position, self.width/2, self.height/2)
    def update_cshape(self):
        self.cshape.center = self.x, self.y
```

Теперь создадим базовый класс по работе со спрайтами (`ActorSprite`). Класс наследуется от класса `Sprite`. Входными параметрами будут: само изображение; его позиция; смещение.

Далее создается ранее не встречавшийся объект движения — `motion`. Здесь снова выступает еще одна новая и сильная сторона движка — это действия (`action`). С помощью `action` можно создавать движение объекта.

В текущем примере используется класс `MoveBy`, его входными параметрами служат координаты смещения и время. В качестве координат указываются не абсолютные координаты, а относительные, то есть в текущем примере спрайт будет смещен на `self.shift` пикселей по абсциссе относительно своего текущего местоположения. Следующий параметр, время, отвечает, за какой промежуток времени будет выполнено это самое перемещение (определяется в секундах).

Затем вызывается функция `do`, которая запускает созданное нами движение, но запускает не просто его, а циклически. Метод `Repeat` как раз отвечает за повторение движения. Но если мы создадим запись по типу `action.Repeat(move)`, то результат не обработает. Спрайт будет двигаться вечно в одном направлении и в конце концов выедет за рамки окна приложения. Во избежание этого мы повторяем то же самое движение, но наоборот; здесь нам помогает метод `Revers`.

Следующий параметр, `self.cshape`, понадобится для учета пересечений (столкновений спрайтов). Вокруг спрайта рисуется невидимый прямоугольник (далее — «рамка»), благодаря которому можно будет определить, было ли пересечение или нет. Так как спрайты на сцене будут в постоянном движении, надо постоянно обновлять невидимую область пересечения, иначе она останется на том месте, где создавался спрайт. Ниже создается функция `update_cshape()`. Именно она позволит невидимой «рамке» двигаться вслед за объектом, путем перемещения центра рамки в центр спрайта.

Настало время создать главный слой сцены, на котором будет происходить все действие:

```
class MainLayer(Layer):
    def __init__(self):
        super(MainLayer, self).__init__()
        self.overlap = False
        self.hero = ActorSprite('data/hero.bmp', (100, 240), 350)
        self.enemy = ActorSprite('data/baddie.bmp', (540, 240), -350)
        self.add(self.hero)
        self.add(self.enemy)
        self.collman = collision.CollisionManagerBruteForce()
        self.collman.add(self.hero)
        self.collman.add(self.enemy)
        self.schedule(self.update)
    def update(self, dt):
        if self.collman.they_collide(self.hero, self.enemy):
            if not self.overlap:
                print "Collision work"
                self.overlap = True
            else: self.overlap = False
        self.hero.update_cshape()
        self.enemy.update_cshape()
```

Класс называется `MainLayer`. В его конструкторе происходит:

- » наследование от родительского класса (`super`);
- » создание логической переменной `overlap`;
- » создание двух спрайтов и затем их добавление на слой (`self.add`) (иначе они не отобразятся);
- » создание менеджера пересечений [`collision`] (`CollisionManagerBruteForce()`).

Менеджер пересечений регистрирует пересечение «рамок», но для начала необходимо ему сообщить, за рамками каких объектов он должен следить. Это выполняется функцией `collman.add()`.

Последняя строчка нашего конструктора класса сообщает `cocos2d`, какую функцию необходимо вызывать каждый кадр (метод `schedule`).

Функция, которая вызывается на каждый кадр, как уже стало понятно, это `update()`. Стоит обратить внимание, что входных параметров два. Первый — это `self`, как обычно принято в Python, а второй — `dt`. Переменная `dt` возвращает количество т.н. «тиков» с момента последнего вызова функции (в нашем случае это `update`), а генерируется `dt` функцией `schedule()`.

В функции `update` происходит проверка на пересечение созданных ранее объектов (`collman.they_collide()`). Если было пересечение, то в консоли будет выведен текст "Collision work".

Далее вызываются методы объектов по обновлению «рамки».

Итак, описание классов закончено. Осталось всё это запустить:

```
director.init(caption = "Test collision")
scene = Scene(MainLayer())
director.run(scene)
```

Теперь можно перейти к реализации игры из выпуска **LXF210**:

```
import random
import sys
from cocos.director import director
from cocos.layer import Layer
from cocos.sprite import Sprite
```

»

```

from cocos.scene import Scene
import cocos.actions as action
import cocos.collision_model as collision
from pyglet.window import key
class BackDrop(Layer):
    def __init__(self):
        super(BackDrop, self).__init__()
        self.backdrop = Sprite('data/backdrop.bmp', (320, 240))
        self.add(self.backdrop)
class ActorSprite(Sprite):
    def __init__(self, image, position):
        super(ActorSprite, self).__init__(image, position)
        self.cshape = collision.AARectShape(position, self.width/2, self.height/2)
    def update_cshape(self):
        self.cshape.center = self.x, self.y

```

Как обычно, программа начинается с импортирования библиотек и создания классов.

Класс `BackDrop` является дочерним классом для `Layer`, и в нем просто инициализируется изображение. Этот слой будет выступать в качестве фона игры.

Класс `ActorSprite` создает спрайт с невидимой рамкой для учета столкновений.

```

class GamePlay(Layer):
    is_event_handler = True
    def __init__(self):
        super(GamePlay, self).__init__()
        self.xscreen, self.yscreen = director.get_window_size()
        self.missile_delay_h = 0
        self.missile_delay_e = 0
        self.missile_list_hero = []
        self.missile_list_enemy = []
        self.enemy_list = []
        enemy_move_x = action.MoveBy((90, 0), 0.5)
        enemy_move_y = action.MoveBy((0, -10), 0.05)
        for i in range(10):
            self.enemy_list.append(ActorSprite('data/baddie.bmp', (50 + i * 50, 430)))
            self.add(self.enemy_list[i])
            self.enemy_list[i].do(action.Repeat(enemy_move_x + enemy_move_y + action.Reverse(enemy_move_x) + enemy_move_y))
        self.hero = ActorSprite('data/hero.bmp', (self.xscreen/2, 80))
        self.add(self.hero)
        self.speed_hero = 5
        self.hero_destroy = False
        self.collman = collision.CollisionManagerBruteForce()
        self.k_right = False
        self.k_left = False
        self.schedule(self.update)

```

Класс `GamePlay` — последний и самый большой в игре. В нем описана вся логика.

Сперва значение переменной класса `is_event_handler` устанавливается в `True`. Этот параметр указывает на необходимость обработки событий, например, нажатия кнопок на клавиатуре и т.п. (по умолчанию — `False`).

В конструкторе класса происходит:

- » определение размеров окна приложения по обеим осям;
- » инициализация переменных для подсчета времени с момента запуска самой последней ракеты для врагов и героя соответственно (`missile_delay`);
- » инициализация списков всех запущенных ракет врагов и героя соответственно (`missile_list`);
- » список всех врагов (`enemy_list`);
- » создание движения по абсциссе и ординате;

- » открытие цикла, в котором создаются десять врагов с добавлением каждого на слой и в общий список (`enemy_list`), а также с установкой движения;
- » создание героя и добавление его на слой;
- » инициализация скорости героя (`speed_hero`) и его статус уничтожения (`hero_destroy`);
- » создание менеджера столкновений;
- » инициализация логических переменных для определения нажатия клавиш;
- » установка функции, которая будет вызываться каждый кадр.

```

    def on_key_press(self, k, modifiers):
        if k == key.RIGHT:
            self.k_right = True
        if k == key.LEFT:
            self.k_left = True
        if k == key.SPACE:
            if self.missile_delay_h == 0 and not self.hero_destroy:
                self.missile_delay_h = 1
                self.__missile_build(self.missile_list_hero, 'data/heromissile.bmp', (self.hero.x, self.hero.y), 400)
    def on_key_release(self, k, modifiers):
        if k == key.RIGHT:
            self.k_right = False
        if k == key.LEFT:
            self.k_left = False

```

Метод `on_key_press` определяет нажатие кнопок на клавиатуре. Если были нажаты клавиши стрелочек влево или вправо, то соответствующие переменные изменяют свое значение на `True`. Если была нажата клавиша пробел, то открывается новое условие, в котором: если время с последнего запуска ракеты равно нулю и сам герой жив — происходят присвоение переменной `missile_delay_h` значения, равного единице, и вызов закрытой функции осуществляющей создание ракеты.

Метод `on_key_release` определяет отпущение кнопок на клавиатуре. Если были отпущены клавиши стрелочек влево или вправо, то соответствующие переменные изменяют свое значение на `False`.

```

    def __count_delay(self, delay, top_mark):
        if delay != 0:
            delay += 1
        if delay == top_mark:
            delay = 0
        return delay
    def __missile_build(self, master_list, source_image, position, shift):
        missile = ActorSprite(source_image, position)
        current_missile = len(master_list)
        master_list.append(missile)
        self.add(master_list[current_missile])
        missile.do(action.MoveBy((0, shift), 2))

```

Закрытый метод `__count_delay` выступает в роли счетчика. Входными параметрами являются текущее значение счетчика и верхняя его граница. Эта функция будет вызываться каждый кадр. С ее помощью происходит задержка способности непрерывно создавать ракеты и врагам, и герою.

- Следующий закрытый метод посвящен созданию ракет. Он способен применяться у обеих враждующих сторон, точно так же, как и метод `__count_delay`. В качестве входных параметров выступают:
- » основной список ракет, в который будет добавляться новая (это текущий список всех ракет героя или ракет врагов);
 - » относительный путь до изображения ракеты;
 - » позиция;
 - » смещение.

В самом методе сначала происходит создание спрайта. Далее находится длина основного списка и добавление в него уже созданного нового спрайта. Добавление ракеты на слой происходит

с помощью уже известной длины списка, так как она будет являться индексом только что добавленного объекта. Дело в том, что если список состоит, к примеру, из двух элементов, то длина этого списка будет равна 2, но индексами соответствующих ячеек будут 0 и 1. Поэтому в данном случае мы сначала находим длину основного списка, затем добавляем нашу ракету и уже потом обращаемся к ней по индексу. Конечно, в Python списки очень гибкие, и эту операцию можно было сделать проще, но сейчас речь не об этом.

Последним действием в этом методе будет установка и запуск движения.

```
def __collision_missile(self, missile_list, crossing_obj):
    for missile in missile_list:
        missile.update_cshape()
        label_coll = False
        if self.collman.they_collide(missile, crossing_obj):
            self.remove(crossing_obj)
            label_coll = True
        if missile.y >= self.yscreen or missile.y <= 0 or label_coll:
            self.remove(missile)
            missile_list.remove(missile)
        if label_coll: return label_coll
```

Метод `__collision_missile()` проверяет столкновение объектов. Входными параметрами являются список ракет и сам объект с которым надо установить факт пересечения.

Для определения пересечения одного объекта с целым списком в начале метода открывается цикл, в котором и перебирается список. В теле цикла происходит обновление невидимой рамки и присвоение ложного значения переменной `label_coll` (служит маркером).

Далее идут три условия:

- 1 Если ракета пересеклась с объектом, то объект удаляется, а значение маркера устанавливается на истинно.
- 2 Если ракета выходит за рамки окна приложения по ординате или было зарегистрировано пересечение, то удаляется сама ракета и ее присутствие в списке.
- 3 Если было зафиксировано пересечение, то сам метод возвращает True.

Стоит отметить, что данный метод является универсальным, то есть его можно использовать и относительно героя, и относительно врагов.

Теперь настало время всё это скомпоновать. Данное действие происходит в методе `update()`:

```
def update(self, dt):
    if len(self.enemy_list) == 0:
        sys.exit()
    for enemy in self.enemy_list:
        enemy.update_cshape()
        label_coll = self.__collision_missile(self.missile_list_hero, enemy)
        if label_coll:
            self.enemy_list.remove(enemy)
        self.hero.update_cshape()
        label_coll = self.__collision_missile(self.missile_list_enemy, self.hero)
        if label_coll:
            sys.exit()
    # перемещение героя
    if self.k_right:
        # проверка окрестности героя
        if (self.hero.x + self.hero.width/2) < self.xscreen:
            self.hero.x += self.speed_hero
    if self.k_left:
        if (self.hero.x - self.hero.width/2) > 0:
            self.hero.x -= self.speed_hero
    self.missile_delay_h = self.__count_delay(self.missile_delay_h, 50)
    if self.missile_delay_e == 0 and len(self.enemy_list):
```

```
self.missile_delay_e += 1
random_enemy = random.choice(self.enemy_list)
self.__missile_build(self.missile_list_enemy, data/
baddiemissile.bmp', (random_enemy.x, random_enemy.y), -430)
self.missile_delay_e = self.__count_delay(self.missile_delay_e, 80)
```

В начале каждой итерации проверяется наличие «живых» врагов в игре. Если их количество равно нулю, то происходит выход из игры.

Затем открывается цикл, в котором перебираются все враги и для каждого определяется наличие столкновение с ракетой героя. Если столкновение было, то враг удаляется.

Далее происходит определение пересечения ракет врагов с героем. Если пересечение было, то осуществляется выход из игры.

Потом происходит реализация движения героя. Если была нажата левая или правая клавиша стрелок на клавиатуре, то будет производиться перемещение героя в соответствующем направлении до тех пор, пока не будет достигнута граница окна приложения по абсциссе или пока не будет отпущена клавиша.

Последнее, что осталось, это использование счетчиков. У героя всё просто и понятно: каждый кадр происходит изменение значения его задержки, если, конечно, оно не равно нулю.

С врагами всё интереснее. За героя стреляет реальный человек, а за врагов стреляет компьютер. Поэтому вначале проверяется, равна ли задержка нулю и есть ли вообще враги на поле. Если оба условия удовлетворены, то происходит изменение параметра задержки врагов на единицу. Следом выбирается случайным образом враг из всего списка врагов и создается ракета в точке расположения, только что выбранной компьютером противника. Благодаря этому, каждая выпущенная ракета врагом будет определяться случайно компьютером.

В конце теперь происходит изменение значения задержки врагов, если, конечно, оно опять же не равно нулю.

```
director.init(caption = "Space Invaders")
scene = Scene(BackDrop(), GamePlay())
director.run(scene)
```

Теперь инициализируем окно и сцену, и запускаем!

Заключение

Движок cocos2d предусматривает большое количество возможностей, выходящих за рамки данной статьи (к примеру, это скелетная анимация, поддержка мозаичных карт, различные графические 3D-эффекты и многое другое). Напоследок упомяну некоторые подводные камни, с которыми я сталкивался:

» **Ужасная документация** Описание методов часто очень скудное, непонятно, как и что использовать, мало примеров. Спасают только демонстрационные программы, доступные в официальном репозитории на GitHub.

» **Рендер мозаичных карт** К сожалению, с ним есть проблема. При использовании достаточно мелкой сетки и максимально возможной заполненной карты могут появиться проблемы с частотой кадров из-за того, что движок не будет справляться со своевременной отрисовкой изображения.

» **Портируемость** По какой-то причине при портировании моего проекта на проприетарную «оконную» систему частота кадров приложения упала в два раза, несмотря на то, что компьютер, на котором изначально всё разрабатывалось под ОС ROSA Fresh, был во много раз слабее компьютера с «оконной» системой.

После долгого исследования *pygame* и *cocos2d* можно прийти к выводу, что эти инструменты отличаются. Каждый из них имеет свои преимущества и недостатки. Сравнить их между собой я не вижу смысла, так как они базируются на разных технологиях и подходы к разработке приложений тоже разные.

С моим проектом, разработанным вне рамок данной статьи, можно ознакомиться по следующей ссылке на GitHub: <https://github.com/RastaDill/GangstaSnowball>. 



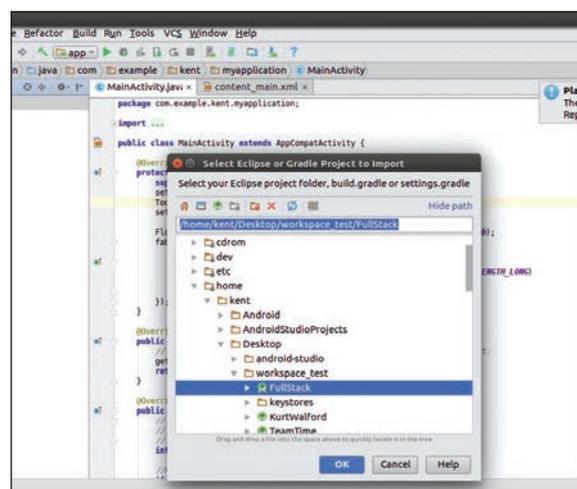
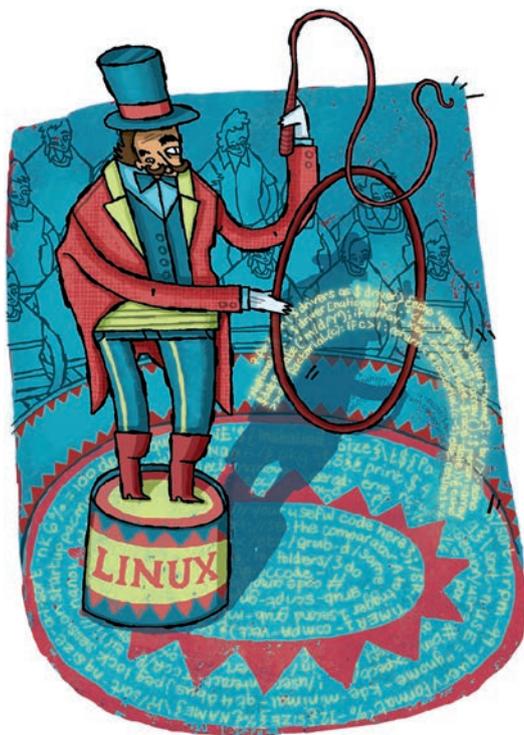
Android Studio: Разрабатываем приложения

Кент Ельчук снимает завесу тайны с «кирпичиков», необходимых, чтобы начать построение приложений для Android и отправить их в Google Play.



Наш эксперт

Кент Ельчук — опытный веб-разработчик, который создал, обезопасил и поддерживает множество веб-приложений для высших школ и колледжей, а также для сайтов.



➤ **Импорт каталога с файлами шаблонов в Android Studio.** Для этого достаточно всего нескольких щелчков мыши.

Возможно, у вас есть устройство Android и вы устанавливали на нем приложения из Google Play. Возможно, вы спрашивали себя: что это за приложения и как они попадают в Google Play. А может, вы даже спрашивали себя: «Как разработать приложение самим?»

На нашем уроке мы раскроем тайну разработки приложений для Android. Вы узнаете, как загрузить и установить Android Studio (она бесплатна), разработать приложение, протестировать приложение в эмуляторах и на настоящем устройстве Android, например, на телефоне Samsung Note, и затем отправить свое приложение в Google Play, чтобы его могли загрузить другие пользователи.

Средств для разработки приложений Android немало, но для нашего урока мы воспользуемся Android Studio, это де-факто основная среда для разработки приложений, и ее можно загрузить со страницы <https://developer.android.com/studio>. Для создания примеров мы воспользуемся языком Java (обычно применяемым для этой цели) и средой Ionic Framework, которая позволяет разрабатывать приложения на основе ныне распространенных web-технологий, таких как HTML, CSS и JavaScript.

Для Android Studio требуется несколько пакетов, но установить их легко. (Мы тестировали свою конфигурацию в 64-битной Ubuntu 16.04 LTS.) Но прежде чем перейти к командам для добавления пакетов, надо включить виртуализацию BIOS, это позволит выполнять тестирование на эмуляторах. Разрабатывать и тестировать приложения можно и без этого, но для тестирования придется подключать свое устройство Android к USB-порту.

Изменить параметры BIOS можно через BIOS Setup во время загрузки. При загрузке вы видите сообщение о том, что для доступа к настройкам надо нажать F9, F1 или F10. На некоторых компьютерах используются другие клавиши. Например, на прекрасных старых ноутбуках Thinkpad с Linux была клавиша 'ThinkVantage', которую можно было нажать для доступа к BIOS после загрузки. Ищите в BIOS параметр 'Intel VT-x', 'Intel Virtualization Technology', 'Virtualization Extensions' или нечто похожее.

После включения параметра в BIOS загрузка продолжится, и вы попадете в обычную систему Linux. Но теперь надо еще раз перезагрузить систему, чтобы включить виртуализацию. Нередко для включения виртуализации следует после изменения параметра в BIOS загрузить компьютер заново.

Покончив с BIOS, можно открыть терминал в Ubuntu и установить KVM. KVM не установится, если виртуализация не включена. С помощью приведенной ниже команды можно установить KVM и проверить, что она работает. Опять же, помните, что если у вас возникли проблемы с KVM или вы не сможете включить

Скорая помощь

Если вы собираетесь использовать Cordova или Ionic Framework, то основной каталог с файлами и подкаталогами, которые вам надо будет изменить — [platforms/android/assets/www](https://github.com/ionic-team/cordova-lib/blob/master/packages/cordova/lib/platforms/android/assets/www).

Изменения текста и кнопок

Наш урок посвящен разработке простого приложения с web-представлением. А если вы хотите создать кнопки, привязанные к внешним источникам? Тогда можно внести в главный класс приложения, файл **Example.java**, описанный в основной статье, некоторые изменения.

Основные изменения кода выполняются в классе **Example**, как показано ниже.

```
public class Example extends Activity {
    Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_test);
        addListenerOnButton();
    }
}
```

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.test, menu);
    return true;
}
public void addListenerOnButton() {
    button = (Button) findViewById(R.id.mybutton);
    button.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View arg0) {Intent
        browserIntent = new Intent(Intent.ACTION_VIEW, Uri.
        parse("https://example.com"));
        startActivity(browserIntent);
        });
    }
}
```

Теперь опишем, что происходит и где нужно внести изменения.

Класс **Example** расширяет класс **Activity** **Example**, и вы видите метод **setContentView()** и переданную ему раскладку. Раскладка расположена в файле **src/res/layout/activity_test.xml**.

Откройте **activity_test.xml**. Внизу вы увидите две вкладки: **Design** [Структура] и **Text** [Текст]. Щелкнув по **Text**, вы увидите раскладку, текстовые представления и кнопки, и сможете выполнить необходимые изменения.

Сравнив кнопки и текстовые ссылки в файле **activity_test.xml**, вы увидите, что в файле не была создана ссылка для кнопки; это было сделано в методе **public void addListenerOnButton()** в классе **Example**.

виртуализацию, вы всё равно можете продолжить работу. Так что если можете — добавьте команду. Если нет, обойдетесь.

```
$ sudo apt-get install qemu-kvm libvirt-bin virtinst bridge-utils
cpu-check
```

```
$ sudo kvm-ok
```

Если KVM установлена правильно, результат будет примерно следующим:

```
INFO: /dev/kvm exists
```

```
KVM acceleration can be used
```

Еще две команды добавляют пакеты для Android Studio, о которых я упоминал чуть ранее (без этих пакетов Android Studio работать не будет):

```
$ sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386
lib32z1 libbz2-1.0:i386
```

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

Загрузив и распаковав Android Studio, можно переместить соответствующий каталог в удобное вам расположение, для простоты задав каталогу новое имя, что-нибудь вроде **android-studio**. Чтобы открыть Android Studio, перейдите в командную строку и зайдите в главный каталог.

Перейти в каталог **android-studio** из текущего расположения можно командой

```
$ cd android-studio
```

Зайдя в этот каталог, войдите на один уровень ниже, зайдите в каталог **bin** и откройте саму Android Studio:

```
$ cd bin
```

```
$ sh ./studio.sh
```

Android Studio откроется и загрузится. С точки зрения web-работчика, это довольно объемная программа. Лично нам показалось, что для ее комфортного использования без провалов по времени требуется жесткий диск SSD и 4 ГБ оперативной памяти.

Чтобы сделать еще один шаг вперед, можно привязать Android Studio к средству запуска, создать образ средства запуска для запуска программы или настроить запуск этой программы при загрузке Ubuntu. Помните, что в вашем дистрибутиве Linux найдутся собственные методы, которые заставят программу вести себя так, как необходимо.

Использование Android Studio

После первого запуска Android Studio вы сможете создать новое приложение с помощью шаблонов по умолчанию или подготовленного пакета. Например, вы могли купить какую-нибудь книгу с программами Android и загрузили несколько файлов для использования в качестве стартовой точки.

В следующем разделе мы объясним, как создать приложение с помощью готового набора кода — обычно, каталога с файлами.

Готовые приложения Android

Один из проектов, включенных в это приложение, предназначен для создания web-представлений, которые по сути представляют собой загрузку сайта, когда пользователь щелкает по иконке приложения.

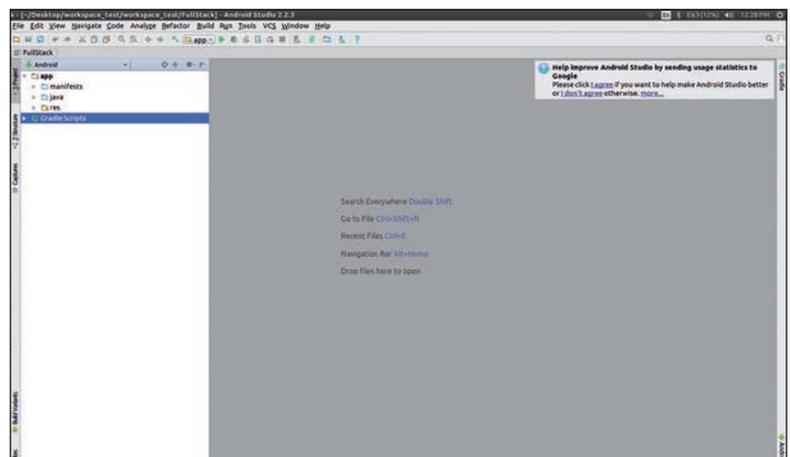
Чтобы начать работу, выберите **File > Import Project > Select Project > Select Folder** [Файл > Импортировать проект > Выбрать проект > Выбрать каталог], затем нажмите **OK** и еще раз **OK**. После этого вы увидите, что Android Studio создает проект **Gradle Project** — с использованием средства сборки **Gradle**.

Если что-то пойдет не так, вы увидите сообщение об ошибке в окне сообщений внизу. Для установки необходимых средств сборки щелкните ссылку в окне сообщений. После каждого обновления достаточно нажать кнопку **Finish** [Готово].

По окончании сборки **Gradle** вам могут предложить обновить **Gradle** до последней версии; вы можете сделать это или пропустить этот шаг.

После загрузки проекта в Android он появится в левой части редактора. Файлы, которые надо изменить, находятся в каталоге **app**. Если их нет, загрузите проект снова.

Файлы, которые вам требуется изменить, находятся преимущественно в каталогах **manifests**, **java** и **res**.



» При первом использовании готового приложения появится пустое окно. Чтобы подстроить приложение для себя, надо изменить несколько ключевых файлов.



Хотя мы стараемся сохранить это упражнение как можно более простым, следует упомянуть, что настанет день, когда вам придется менять файл `build.gradle` в каталоге со скриптами Gradle, для настройки параметров `minSdkVersion` и `targetSdkVersion`. С версией SDK сейчас заморочиваться незачем, но мы поясним общий смысл эти терминов. Представьте себе старый и новый телефоны Android. На старом используется более старая версия SDK, а на новом — SDK с более высоким номером версии. Так вот, низкое значение `minSdkVersion` позволяет обеспечить обратную совместимость.

В `targetSdkVersion` надо задать последнюю версию API. На момент написания статьи это 25. При этом вы всегда можете внести изменения в файл и пересобрать пакет APK. Собрать пакет просто: достаточно выбрать `Build > Build APK` [Сборка > Собрать пакет APK] в главном меню.

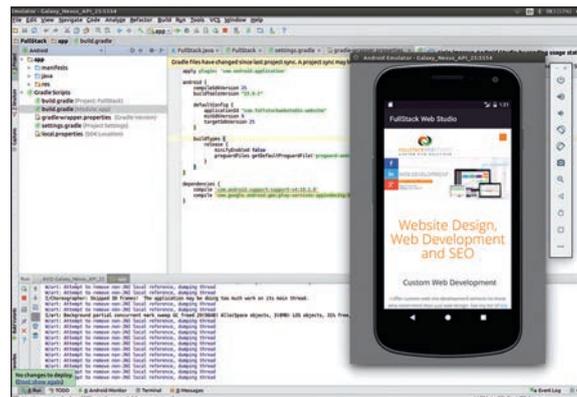
Довольно об APK. Давайте перейдем к делу и протестируем пакет. Без устройства Android или эмулятора никакое тестирование вашего приложения невозможно. Быстрее всего будет взять телефон Android и подключить его к компьютеру через порт USB.

Телефон, поставляемый без кабеля, представить сложно. Если у вас нет кабеля или вы потеряли его, придется купить новый кабель. Разъемы у телефонов разные, поэтому захватите с собой телефон в магазин, чтобы не ошибиться с кабелем.

Вы сможете протестировать приложение сразу же после подключения телефона к компьютеру. Для тестирования можно нажать зеленую стрелку вправо на панели инструментов под главным меню. Также можно выбрать `Run > Run` [Запуск > Запустить].

Откроется окно `Select Deployment Target` [Выбор цели развертывания]. В верхней части окна вы увидите список `Connected Devices` [Подключенные устройства] с вашим устройством. Если его нет, взгляните, что происходит на вашем устройстве: там может появиться приглашение для подключения, на которое надо отреагировать.

Если забежать немного вперед, в этом списке также будут показаны все программные эмуляторы, которые вы создадите. Пока просто выберите свое устройство и нажмите кнопку `OK`. Может появиться окно `Instant Run` [Немедленный запуск] с вариантами действий, например, `Install and Continue` [Установить и продолжить]. При нажатии на эту кнопку будут установлены необходимые компоненты с сайта Google.com.



➤ Эмулятор в работе выглядит как настоящий телефон и позволяет тестировать приложение даже без физического устройства.

По окончании установки (при необходимости) Gradle сделает свое дело, и вам останется только подождать, пока будут выполняться сборка и индексация.

Если после этого снова запустить приложение, вы должны увидеть вариант для использования своего устройства. Если нет, размонтируйте и повторно смонтируйте устройство, используя другой порт USB. Затем запустите приложение снова. И еще кое-что: так как в приложении загружается `web-представление`, убедитесь, что телефон подключен к Интернету.

Если не углубляться в детали, то `web-представление` может само загрузить браузер; или оно может выглядеть как обычное приложение, особенно если сайт запрограммирован хорошо и быстро загружается. `Web-представление` к этому уроку будет загружаться так, что оно будет выглядеть как настоящее приложение, поскольку панель браузера придаст бы ему нежелательный вид.

Код к этому уроку взят из персонального проекта, и его можно тестировать, не обижая автора. Однако вы явно захотите изменить этот проект для себя, и сейчас мы объясним, как это сделать.

Поскольку теперь всё работает так, как ожидалось, советуем сделать копию проекта, так как новичку и даже опытному программисту легко совершить ошибки, трудно диагностируемые на основе полученных результатов. Те, кто уже имеет некоторый опыт, знают, как это бывает: скопируйте сообщение об ошибке в Google, и либо первый приличный ответ решит проблему, либо вы будете часами хвататься за соломинки, пытаясь нарвать подходящий ответ в куче тех, которые относятся к другим версиям.

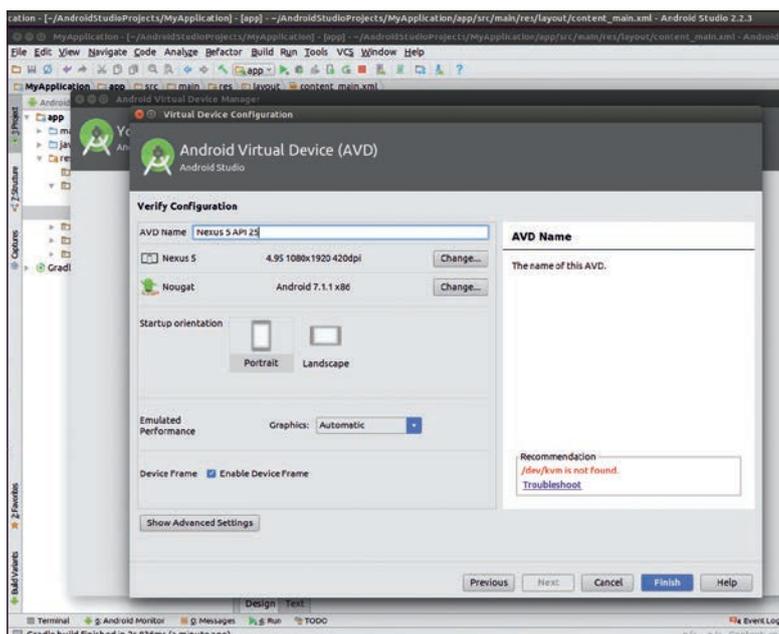
Еще одно предупреждение о проблемах с Android Studio: любые изменения в коде автоматически сохраняются. Помните это, особенно если у вас не было опыта работы с такими редакторами.

В Android Studio, как и в некоторых других редакторах, используется *IntelliJ Idea*. Те из вас, кто пользовался похожими программами от JetBrains, такими как *PhpStorm* или *WebStorm*, почувствуют себя как дома.

В главном меню редактора также есть вкладка `VCS`. При желании можно воспользоваться системами управления версиями, такими как *Git* или *Subversion*. Кроме того, у редактора есть функция локальной истории, которая сама по себе является облегченной системой для управления версиями. При изменениях в файле в разделе `Local History` [Локальная история] вы увидите время изменения файлов.

Теперь, когда мы прояснили основные моменты в работе редактора, вернемся к нашему приложению и сделаем так, чтобы в нем запускалось выбранное `web-представление`. Уточню, что сама `web-страница` создается в виде одностраничного приложения с использованием `AngularJS`, она быстро загружается и позволяет выполнять быструю навигацию по содержимому.

Сперва разберемся с правилами именования при отправке приложения в `Google Play`. Важно, чтобы имя пакета представляло



➤ Вы можете добавить эмуляторы, если у вас включены KVM и виртуализация. Эмуляторы обычно работают медленнее устройства.

Cordova и Ionic Framework

У многих из тех, кто начинает заниматься разработкой приложений, уже есть знания HTML, CSS и JavaScript, популярных клиентских технологий для создания сайтов.

Чтобы приступить к разработке, надо выполнить несколько действий, и прежде всего — открыть терминал.

Следует установить Node.js, диспетчер пакетов Node.js, Apache Cordova и (при желании) Ionic Framework.

```
sudo apt-get update
sudo apt-get install nodejs
sudo apt-get install npm
sudo npm install -g cordova
sudo npm install -g ionic
```

Установив необходимые пакеты, перейдем к настройке и редактированию. Первым делом создадим новый проект Apache Cordova. Для этого

перейдите в каталог, в котором будет размещаться пакет. Для простого примера можно использовать каталог Desktop:

```
cd Desktop
mkdir cordova
cd cordova
Теперь можно запустить новый проект. Перед этим скорее всего потребуются создать символическую ссылку на Node.js из Node, так как Cordova может выдать ошибку, пойдя по неправильному пути. Для этого выполните следующие команды:
ln -s /usr/bin/nodejs /usr/bin/node
cordova create example com.example.hello HelloWorld
После этого сделайте следующее:
cd example
cordova platform add android --save
```

Дополнительную информацию см. на <https://cordova.apache.org/docs/en/latest/guide/cli/>.

Запустим новое приложение с помощью Ionic Framework:

```
cd
cd Desktop
mkdir ionic
ionic start example blank
cd example
ionic platform add android
```

Чтобы собрать и протестировать приложение Cordova, выполните следующие команды:

```
cordova build
cordova run android
Для Ionic команды будут такими:
ionic build android
ionic run android
```

Дополнительные подробности и справку вы найдете на <http://ionicframework.com/docs/guide/starting.html>.

собой «перевернутое» имя сайта, к которому подключено web-представление. Т.е. если у вас есть web-представление для сайта **example.com**, имя пакета будет представлять собой что-то вроде **com.example.website**. Как вы видите, оно начинается с 'com.example', а не с 'example.com'. Это простое правило.

Так как в этом примере пакета выполняется перенаправление на web-представление, можно выполнить полный поиск по пути для доменного имени во всех файлах. Для этого щелкните на папке приложения правой кнопкой мыши, выберите Find In Path [Поиск по пути] и введите строку для URL-адреса вашего сайта. Например, для пакета **com.example.website** введите в строке поиска 'example'.

Вы увидите множество ссылок в результатах поиска. Вскоре мы покажем, как изменить ссылки сразу во всех файлах.

Для приложений Android название пакета содержится в файле **app/java/com.example.website/example**. Название пакета — это «сердце» проекта.

Так как в этом упражнении мы создаем шаблон для вашего проекта, сначала надо сделать копию существующего рабочего приложения, поскольку оно у нас уже есть.

Чтобы переименовать проект, его надо сначала закрыть. Затем открывайте каталог приложения и переименовывайте файлы и каталоги и немного меняйте текст. Если тщательно следовать инструкциям, этот процесс займет всего минуту.

Откройте каталог приложения Android (тот самый каталог, в который вы изначально поместили приложение).

Перейдите в файл **app/src/main/AndroidManifest.xml** и измените приведенную ниже строку на желаемый URL. Так, **com.example.com** можно изменить на **com.mywebsite.website**:

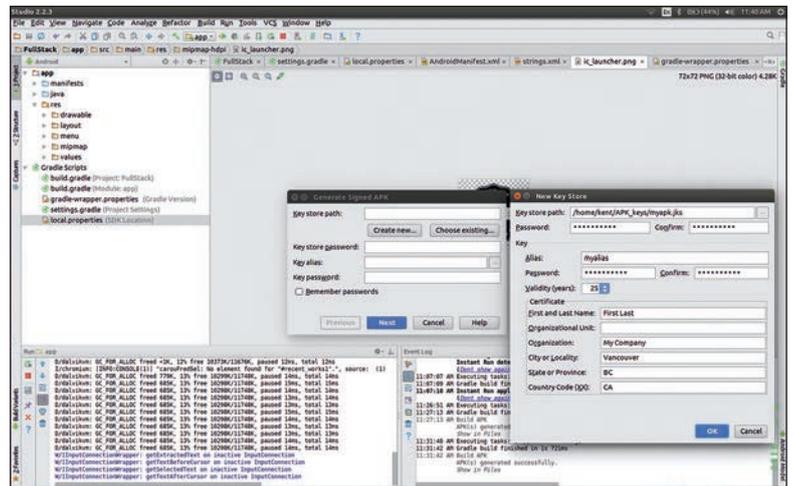
```
package="com.example.website"
```

После этого откройте файл **app/build.gradle** и укажите в следующей строке тот же URL, который указали в файле **AndroidManifest.xml**.

```
applicationId "com.example.website"
```

Затем перейдите в каталог **app/src/main/java/com** и переименуйте его в соответствии с именем пакета. Например, с учетом наших предыдущих изменений каталог надо назвать **mywebsite**.

Теперь откройте приложение Android в Android Studio. Перейдите к каталогу **app/java/com.mywebsite.website** и переименуйте файл в этом каталоге. В данном случае файл будет назван именем **Mywebsite**.



► **Перед отправкой приложения в Google Play потребуется подписанный пакет APK. Также придется заполнить несколько полей.**

При переименовании этого файла вы увидите, что уже проставлено множество галочек — это Android Studio помогает выполнить необходимые изменения, например, изменить имя класса и другие важные параметры.

Теперь можно выбрать **Build > Clean and Build > Rebuild Project** [Сборка > Очистка и сборка > Собрать проект]. Запустите приложение, и теперь у вас есть собственный пакет. Поздравляем!

Однако давайте не будем слишком забегать вперед, ведь нам еще надо изменить это представление, включив в него свое содержимое. Это очень просто. Откройте файл **app/java/com.mywebsite.website/Mywebsite**, найдите строку с URL и измените его на адрес сайта, который хотите открыть.

Этот файл позволит вам получить представление о языке программирования Java, используемом для создания «родных» приложений Java. Данный конкретный код используется для отображения web-представления, но его можно модифицировать с целью отображения кнопок и содержимого самого приложения, а не web-представления.

При загрузке приложения вы увидите, что готовый заголовок вам не подойдет. Чтобы изменить его, перейдите в файл

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



AndroidManifest.xml file и измените метки Android. Метки Android можно жестко закодировать или оформить как ссылки на файл **/app/res/values/strings.xml**.

В нашем примере при загрузке приложения на тестовом устройстве его заголовок берется из файла **strings.xml** (строка 'app_name'). Внесите свои изменения.

Теперь перейдем к изображениям. Вам потребуется добавить несколько изображений, которые будут использоваться для нескольких экземпляров. Изображения расположены в двух каталогах: **app/res/drawable** и **app/res/mipmap**. Эти папки удобно смотреть в обычном средстве просмотра каталогов, а не в Android Studio.

Открыв каталог **appfoldername/app/src/main/res/**, вы увидите множество каталогов, имена которых начинаются с 'mipmap' и 'drawable'. Для получения полной копии всех mipmap можно перейти на сайт <https://romannurik.github.io/AndroidAssetStudio/> и сделать это за несколько секунд. Подставив 'drawable', можно сделать то же самое.

Вот, по сути, и все действия по созданию простого приложения. Осталось подготовить его для отправки в Google Play.

Для начала выберите в главном меню **Build > Build APK** [Сборка > Собрать пакет APK]. Этот пакет APK будет расположен в каталоге **app/build/outputs/apk**.

Следующий шаг — подписать пакет APK. Для этого выберите **Build > Generate Signed APK** [Сборка > Создать подписанный пакет APK]. Этот процесс не требует пояснений. Вам понадобится выбрать каталог для сохранения ключей на компьютере.

Если всё хорошо, нажмите кнопку **Next** [Далее]. Убедитесь, что в качестве типа сборки указан **Release** [Выпуск], и нажмите кнопку **Finish** [Готово].

После этого вы должны увидеть всплывающее окно, подтверждающее отсутствие ошибок и предупреждений. Файл APK, который нужно отправить Google Developer, расположен в каталоге **app**.

Теперь загрузите этот файл в Google Developer и подождите несколько часов или даже дней, прежде чем он будет одобрен.

На нашем уроке мы объяснили, как создать простое приложение и подготовить его для Google Play. Наконец, пора описать процесс отправки приложения в Google Play. Первый шаг — зарегистрироваться и заплатить сбор в \$25 с использованием учетной записи Gmail. После этого остальные действия по отправке приложения выполняются очень быстро.

Заплатив сбор и войдя в учетную запись Google Play Developer, выберите **Create Application** [Создать приложение] и укажите его имя. Дальнейшие действия будут понятны сами собой.

В меню слева появится список разделов сверху вниз. По завершении каждого раздела в столбце слева от этого раздела появится зеленая стрелка. В процессе отправки участвуют поля, которые необходимо заполнить. Иногда они не бросаются в глаза, поэтому обязательно обращайтесь внимание на звездочки, отмечающие обязательные поля.

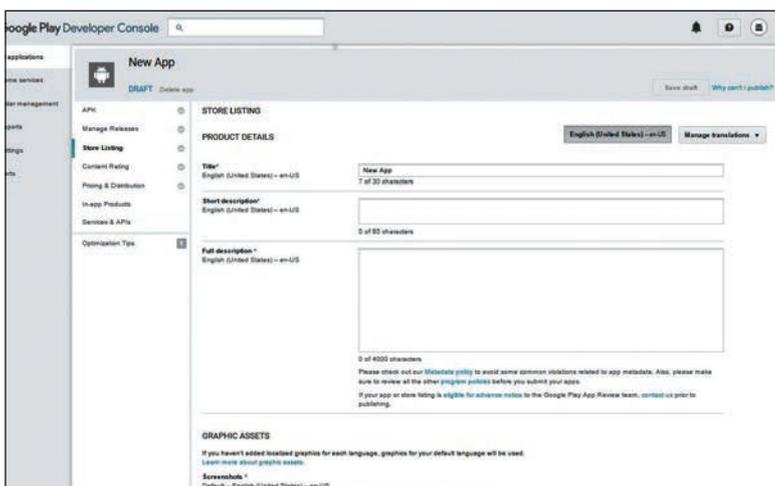
Подготовьте для приложения несколько изображений для различных размеров экрана. Также учтите, что в некоторые поля можно ввести большой объем текста, это позволит лучше продемонстрировать ваше приложение.

Кроме того, помните, что до перевода приложения в рабочий режим вы можете его протестировать, для этого может потребоваться открыть к нему доступ для определенных пользователей.

Наконец, обязательно отследите, где сохранится подписанный файл APK, потому что именно этот файл вы отправите в Google Play.

Заключение

Итак, вы получили пошаговое руководство по созданию приложения Android с помощью Android Studio в Linux. По сравнению с другими операционными системами, в Linux это можно сделать быстро, и всё будет сразу работать без ошибок. **LXF**



➤ Заплатив небольшой взнос Google Play Developer, вы сможете отправить свое приложение на утверждение.

Сборка и тестирование с Cordova и Ionic

При возникновении проблем обычно их можно решить, кое-что поправив в файле **.bashrc** или обновив Java.

Для новичков: если путь в переменной **ANDROID_HOME** не установлен, можно найти его в Android Studio. Загрузите приложение и определите путь, выбрав **Tool > Android > SDK Manager** [Инструменты > Android > Диспетчер SDK]. Вверху вы увидите путь, который надо записать в переменную **ANDROID_HOME**.

В Java можно установить JDK Java из Oracle. Для этого воспользуйтесь несложными инструкциями на www.java.com/en/download/help/linux_x64_install.xml. В данном случае был создан каталог **/usr/java**, и загруженный файл tar был перемещен в этот

каталог. После распаковки был создан новый каталог **jdk1.8.0_121**. Этот каталог станет переменной **JAVA_HOME**. Вы, вероятно, захотите удалить предыдущие версии, которые могут находиться в каталоге **/usr/lib/jvm/**.

```
export JAVA_HOME=/usr/java/jdk1.8.0_121'  
export ANDROID_HOME=/home/myusername/  
Android/Sdk'  
export PATH=${PATH}:${ANDROID_HOME/  
platform-tools}:${ANDROID_HOME/tools
```

Изменив файл **.bashrc**, можете выполнить команду **source ~/.bashrc**, открыть терминал и проверить, что пути установлены, командой

```
echo $JAVA_HOME  
echo $ANDROID_HOME
```

Теперь вы сможете выполнить сборку и снова запустить программу с помощью упомянутых выше команд.

Чтобы загрузить проект в Android Studio, выберите **File > Open > Select App folder > Platforms > Android** [Файл > Открыть > Выбрать каталог приложения > Платформы > Android]. Затем Android проведет вас по процессу установки. Обязательно выберите SDK, когда вам будет предложено это сделать.

После этого можно проверять приложение Cordova или Ionic точно так же, как и обычное приложение. Следует отметить, что в Cordova позволяет добавлять плагины для различных функций, таких как геолокация и хранение данных.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Новое поколение средств защиты

Межсетевые экраны ССПТ, не имеющие IP-адреса

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

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

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

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

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

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

Дополнительное устройство защиты для:

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

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

Python: Личная утилита `find`

Утилита `find` очень удобна для поиска файлов и каталогов. **Михалис Цукалос** покажет, как написать свою версию этой утилиты.



Наш эксперт

Михалис Цукалос (@mactsouk) имеет степень магистра в информационных технологиях от Университетского колледжа Лондона и степень бакалавра по математике. Он администрирует базы данных, пишет программы и работает в Unix как сущая математическая машина.

Из этого урока вы узнаете, как реализовать основные функции утилиты командной строки `find` на Python 3. `Find` — исключительно популярный и удобный инструмент Unix, который помогает вам находить файлы и каталоги на компьютере с Linux.

На нашем уроке мы реализуем лишь небольшую часть утилиты `find`. Но, разобравшись в представленном коде на Python 3, вы сможете сами добавить необходимые функции.

Основы `find`

В этом разделе мы проиллюстрируем использование утилиты `find` на простых примерах.

Вот простой вариант использования `find`:

```
$ find /home/mtsouk -name myFind.py
/home/mtsouk/docs/myFind.py
$ find . -name python
./python
./python/python
$ find . -type d -name python
./python
$ find . -type -name python
find: Arguments to -type should contain only one letter [Опции
-type должны содержать только одну букву]
```

Первая команда ищет во всех каталогах с именами, начинающимися с `/home/mtsouk`, файлы или каталоги с точным именем `myFind.py`. Путь `/home/mtsouk` — абсолютный, так как он начинается с корневого каталога (`/`). Если файлы или каталоги найдутся, пути к ним будут выведены на экран. В противном случае, в соответствии с философией Unix, утилита не выведет ничего. Если найдено несколько файлов или каталогов, будут выведены все они.

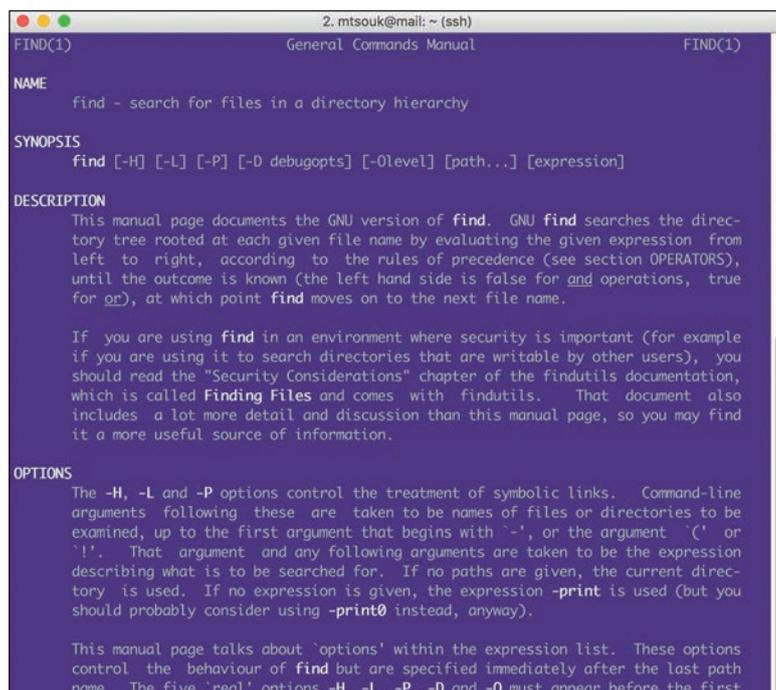
Вторая команда `find` просматривает дерево каталогов, которое начинается с текущего каталога, в поиске файлов или каталогов с именем `"python"`. Третья команда ищет только каталоги. В последнем примере приведено сообщение об ошибке из-за того, что пропущены опции параметра `-type`.

Учтите, что в системах Unix точка (`.`) указывает на текущий рабочий каталог, а две точки (`..`) — на родительский каталог. Это очень удобно, если вы не хотите указывать полные пути.

Утилита `find` очень мощная и умеет многое, но на нашем уроке все ее возможности не охватить. Однако, создав версию `find` на Python 3, ее довольно легко расширить для поддержки новых функций. Сложнее всего создать первую рабочую версию.

Знать, что вы собираетесь реализовать, очень важно, поэтому всегда старайтесь четко определить, какие функции должна иметь ваша программа!

На рисунке слева показана небольшая часть `man`-страницы утилиты `find`.



➤ На рисунке показана небольшая часть `man`-страницы для утилиты `find`. Чтобы просмотреть ее, наберите `man find`.

`__main__` и `__name__`

В скриптах Python 3 рекомендуется (хотя это и не обязательно) начинать выполнение с проверки значения переменной `__name__`. Взгляните на следующий код, который сохранен в файле `b.py`:

```
if __name__ == '__main__':
    print('This is b.py executed as', __name__)
else:
    print('This is b.py executed from:', __name__)
```

Переменная `__name__` необходима для проверки, был ли файл импортирован из другого модуля. Если скрипт Python 3 запускается как автономная программа, значение переменной `__name__` таково: `__main__`. Это значение определяется в Python 3 автоматически. При запуске приведенного выше кода мы получим следующий результат:

```
$ cat a.py
import b
print("Это модуль a!")
$ python3 b.py
Это b.py выполненный как __main__
$ python3 a.py
Это b.py выполненный из: b
Это модуль a!
$ ls -l __pycache__
total 4
-rw-r--r-- 1 mtsouk mtsouk 267 Oct 15 15:40 b.cpython-34.pyc
```

Утилита командной строки `time`

Утилита командной строки `time` поможет вам оценить производительность программы, предоставив полезную информацию о ее выполнении. Эта утилита запускает указанную команду и выдает статистику ее использования после того, как команда завершит работу.

Пример вывода команды `time` показан на рисунке справа.

Временная статистика состоит из трех частей: реального времени, пользовательского времени и системного времени. Реальное время (`real`) — это фактическое время от начала до окончания. Пользовательское время (`user`) — это время, потраченное центральным процессором на код пользовательского режима. Системное время (`sys`) — это время, потраченное в ядре. Проще говоря, системное время — это время, пока ваша программа ожидала своей очереди в Linux, а пользовательское время — это время, в течение которого ваша программа занимала центральный процессор. Если сложить пользовательское и системное время, вы получите общий объем использованного процессорного времени. Если на вашем компьютере несколько процессоров и в вашем процессе используется несколько потоков, то эта сумма может превышать значение реального времени.

```
iMac:code mtsouk$ time python3 testSpeed.py /var
Visited 6683 files!

real    0m0.049s
user    0m0.028s
sys     0m0.017s
iMac:code mtsouk$ time python3 testSpeed.py /var
('Visited', 6683, 'files!')

real    0m0.091s
user    0m0.034s
sys     0m0.051s
iMac:code mtsouk$ time python3 testSpeed.py /usr
Visited 135291 files!

real    0m0.383s
user    0m0.171s
sys     0m0.208s
iMac:code mtsouk$ time python3 testSpeed.py /usr
('Visited', 135291, 'files!')

real    0m0.975s
user    0m0.364s
sys     0m0.582s
```

➤ На рисунке показано различие производительности между версиями `os.walk()` в Python 2.7.10 и Python 3.5.2.

```
$ file __pycache__/_b.cpython-34.pyc
```

```
__pycache__/_b.cpython-34.pyc: python 3.4 byte-compiled
```

Воспользовавшись `b.py` в качестве модуля, вы можете обнаружить в текущем каталоге новый подкаталог, `__pycache__`. Каталог `__pycache__` содержит байт-код Python 3, который используется для ускорения программ Python 3. При желании этот каталог можно и удалить, но когда вы в следующий раз воспользуетесь модулем `b.py` или любым другим модулем, этот каталог будет создан заново.

`file` — это команда Linux, с помощью которой можно определить тип файла.

Таким образом, назначение переменной `__name__` состоит в том, чтобы вы узнали, запущен ли код Python в качестве автономной программы или нет, и выполнить соответствующие действия. Из-за ограниченного объема урока мы будем использовать этот метод только в окончательной версии кода.

Прогуляемся по `os.walk()`

Метод `os.walk()` очень удобен для рекурсивного просмотра и обработки всех файлов и каталогов в дереве каталога, начиная с указанного корневого каталога. Очень интересно отметить, что метод `os.walk()` использует только один параметр, имя каталога, в который он должен зайти. Затем цикл `for` делает всю работу, перебирая все подкаталоги и файлы в корневом каталоге, который был указан в качестве параметра `os.walk()`. В следующем коде, который сохранен в файле `learnWalk.py`, иллюстрируется использование метода `os.walk()`:

```
#!/usr/bin/env python3
```

```
import os
import sys
```

```
if len(sys.argv) >= 2:
    directory = str(sys.argv[1])
else:
    print('Не хватает аргументов!')
    sys.exit(0)
```

```
for root, dirs, files in os.walk(directory):
```

```
    print('**', root)
```

```
    for file in files:
```

```
        pathname = os.path.join(root,file)
```

```
        if os.path.exists(pathname):
```

```
            print(pathname)
```

Запустив файл `learnWalk.py`, мы получим следующий результат (для экономии места мы сократили его):

```
$ ./learnWalk.py ~/code
```

```
** /home/mtsouk/code
```

```
/home/mtsouk/code/a.out
```

```
** /home/mtsouk/code/Haskell
```

```
...
```

```
/home/mtsouk/code/C/sysProg/sparse.c
```

```
/home/mtsouk/code/C/sysProg/filetype.c
```

Как видите, скрипт `learnWalk.py` заходит во все каталоги, начиная с корневого, и выводит на экране пути ко всем найденным файлам или каталогам. Это просто, но очень эффективно. Метод `os.path.exists()` проверяет, что файл существует, прежде чем вывести его имя.

Прежде чем продолжить, убедитесь, что вы полностью разобрались в этом методе.

Теперь пора представить первую версию реализации утилиты `find` для Python 3.

Поиск в Python 3

Теперь, когда у нас есть все необходимые фрагменты кода, настало время объединить их и создать первую версию утилиты `find`, которая будет называться `firstFind.py`. Эта версия принимает два аргумента командной строки: первый — имя каталога, с которого начнется поиск, а второй — имя файла для поиска.

Содержимое файла `firstFind.py` таково:

```
#!/usr/bin/env python3
```

```
import os
import sys
```

```
if len(sys.argv) >= 3:
```



Код к этому уроку можно загрузить с www.linuxformat.com/files/code/tms16.code.zip.

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



```

2. mtsouk@mail: ~/code (ssh)
mtsouk@mail:~/code$ ./myFind.py . doesNotExist
mtsouk@mail:~/code$ ./myFind.py . swap
./C/cUNL/swap
mtsouk@mail:~/code$ ./myFind.py . a.out
./a.out
./C/a.out
mtsouk@mail:~/code$ ./myFind.py . python
./python
./python/python
mtsouk@mail:~/code$ ./myFind.py . a.out -d
mtsouk@mail:~/code$ ./myFind.py . a.out -f
./a.out
./C/a.out
mtsouk@mail:~/code$ ./myFind.py . python -f
./python/python
mtsouk@mail:~/code$ ./myFind.py . python -d -f
./python
./python/python
mtsouk@mail:~/code$ ./myFind.py
Usage: ./myFind.py directory filename [-df]
mtsouk@mail:~/code$

```

➤ На рисунке показаны примеры запуска myFind.py, в основном для проверки скрипта.

```

directory = str(sys.argv[1])
filename = str(sys.argv[2])
else:
    print('Не хватает аргументов!')
    sys.exit(0)

for root, dirs, files in os.walk(directory):
    print("**", root)
    for file in files:
        pathname = os.path.join(root,file)
        if os.path.exists(pathname):
            if file == filename:
                print(pathname)

```

При запуске файла **firstFind.py** мы получим следующий результат:

```

$ ./firstFind.py . a.out
** .
./a.out
** ./Haskell
** ./python
** ./perl
** ./C
./C/a.out
** ./C/system
** ./C/example
** ./C/cUNL
** ./C/sysProg
$ ./firstFind.py
Не хватает аргументов!

```

С помощью приведенного результата очень легко убедиться, что программа **firstFind.py** действительно работает. Записи, начинающиеся с **, представляют собой каталоги — таким образом мы показываем, что программа **firstFind.py** зашла во все нужные каталоги. В окончательной версии нашей утилиты такого вывода уже не будет. С помощью **firstFind.py** также можно найти необходимые файлы.

Пока похоже, что **firstFind.py** работает как ожидалось, поэтому мы можем продолжить и реализовать недостающие функции, которые должна иметь простая команда **find**. Самая важная из недостающих функций — поддержка параметров командной строки, и это, возможно, будет не так уж просто.

Окончательный вариант

Окончательная версия **find** называется **myFind.py**. Ее основное отличие от **firstFind.py** заключается в том, что **myFind.py** принимает два параметра (ключа) командной строки. Параметр **-d** означает, что надо искать только каталоги, а **-f** — что надо искать только файлы. Если указать оба параметра или не указать ни одного, **myFind.py** будет искать и то, и другое!

Первым аргументом командной строки обязательно должно быть имя каталога, а вторым — имя файла или каталога, который вы хотите найти. Поэтому наши два параметра указываются в конце команды в произвольном порядке.

Код файла **myFind.py** таков:

```

#!/usr/bin/env python3
import os
import sys

def find(directory, filename, dirOnly, fileOnly):
    if (dirOnly == 0 and fileOnly == 0):
        dirOnly = 1;
        fileOnly = 1;

    for root, dirs, files in os.walk(directory):
        if dirOnly == 1:
            if os.path.basename(os.path.normpath((root))) == filename:
                print(root)
        for file in files:
            pathname = os.path.join(root,file)
            if os.path.exists(pathname):
                if fileOnly == 1:
                    if file == filename:
                        print(pathname)

def main():
    dirOnly = 0;
    fileOnly = 0;
    if len(sys.argv) == 3:
        directory = str(sys.argv[1])
        filename = str(sys.argv[2])
    elif len(sys.argv) == 4:
        directory = str(sys.argv[1])
        filename = str(sys.argv[2])
        option1 = str(sys.argv[3])
        if option1 == "-d":
            dirOnly = 1;
        if option1 == "-f":
            fileOnly = 1;
    elif len(sys.argv) >= 5:
        directory = str(sys.argv[1])
        filename = str(sys.argv[2])
        option1 = str(sys.argv[3])
        option2 = str(sys.argv[4])
        if (option1 == "-d" or option2 == "-d"):
            dirOnly = 1;
        if (option1 == "-f" or option2 == "-f"):
            fileOnly = 1;
    else:
        print('Использование: ', sys.argv[0], 'directory filename [-df]')
        sys.exit(0)
    # If the given path exists do your job
    if os.path.isdir(directory):
        find(directory, filename, dirOnly, fileOnly)
    else:

```

➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

```
print('Каталог ', directory, ' не существует!')

if __name__ == '__main__':
    main()
else:
    print('Это отдельная программа, а не модуль!')
```

Как видите, переменная `__name__` обычно используется вместе с функцией `main()`; если вы захотите использовать `myFind.py` в качестве модуля, наличие функции `main()` позволит использовать функции `myFind.py` в других программах.

И, как видите, немалая часть функции `main()` посвящена обработке параметров командной строки. Хотя существуют модули, которые позволяют обращаться с аргументами и ключами с меньшим количеством кода, в нашей реализации мы используем другую схему для простоты понимания кода. Основные функции программы реализованы в функции `find()`, которая более или менее похожа на код в файле `firstFind.py`.

Функция `find()` также следит за тем, чтобы отображались только требуемые типы файлов. Вызовы методов `os.path.basename()` и `os.path.normpath()` нужны для извлечения последней части пути, которая представляет собой имя каталога, и для ее сравнения с именем файла, который вы хотите найти.

Это вызвано тем, что `myFind.py` не может корректно сравнить строку вроде `"code"` с полным путем, например, таким как `/home/mtsouk/code`.

На другом уроке этой серии мы подробнее поговорим о том, как обрабатывать в Python 3 аргументы и параметры командной строки.

Тестирование кода

Тестирование — очень важная часть разработки на любом языке программирования, особенно если ваш скрипт работает с системными файлами и каталогами. Поэтому вам придется выполнить ряд тестов для проверки, что скрипт `myFind.py` работает как ожидалось — без тестирования утилиту нельзя разворачивать в рабочей системе, так как это может подвергнуть риску ее надежность и стабильность.

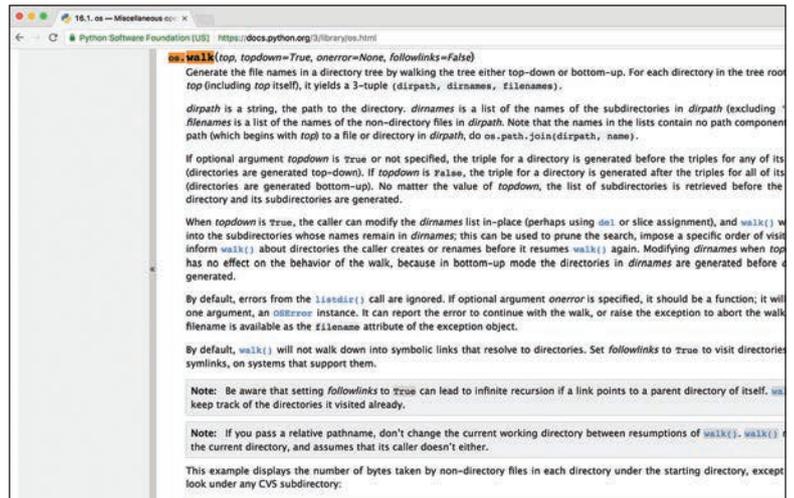
В первом тесте мы попробуем найти файл, которого не существует, во втором — найти файл или каталог, который существует в одном экземпляре, а в третьем — найти файл, который существует несколько раз в анализируемой структуре каталогов. Так как один и тот же файл не может находиться несколько раз в одном и том же каталоге, в третьем случае один и тот же файл будет находиться в разных каталогах.

В последних двух тестах мы проверим, что скрипт `myFind.py` умеет различать файлы и каталоги при указании соответствующих параметров. На рисунке в верхней части стр. 90 показано выполнение этих тестов со скриптом `myFind.py`.

Хорошая новость в том, что тестируя код, вы также учитесь, как его применять.

При запуске скрипт `myFind.py` выдает следующее вспомогательное сообщение:

```
$ ./myFind.py
Использование: ./myFind.py directory filename [-df]
```



➤ На этом скриншоте показана большая часть онлайн-документации для метода `os.walk()`, реализующего базовую функциональность скрипта `myFind.py`.

Сравнение os.walk() и os.scandir()

Если у вас Python версии 3.5 или новее, можете вместо `os.walk()` использовать функцию `os.scandir()`, так как `os.scandir()` намного быстрее `os.walk()` благодаря отсутствию вызовов `os.stat()`. Хорошие новости в том, что если использовать Python 3.5 или более поздней версии, то `os.walk()` будет автоматически использовать реализацию `os.scandir()`, т.е. вам не придется переписывать код.

В последней части урока мы измерим различие в скорости между `os.walk()` и `os.scandir()`, воспользовавшись сравнительно простой командой `time`. Мы проверим версию Python 3.5.2, в которой `os.scandir()` есть, и Python 2.7.10, без `os.scandir()`. Чтобы проверка была максимально достоверной, поиск надо производить по большому дереву каталогов, например, `/usr` или `/var`. В этом разделе мы воспользуемся измененной версией скрипта `learnWalk.py`, которая сохранена в файле `testSpeed.py` и содержит следующую реализацию:

```
import os
import sys

if len(sys.argv) >= 2:
    directory = str(sys.argv[1])
else:
    print('Не хватает аргументов!')
    sys.exit(0)

total = 0
for root, dirs, files in os.walk(directory):
    for file in files:
        total = total + 1
print('Просмотрено', total, 'файлов!')
```

Результаты демонстрируют различие во времени для двух слегка различающихся реализаций — версия для Python 3.5 гораздо быстрее! **LXF**

Поиск документации для os.walk()

На нашем уроке мы рассматриваем относительно сложные методы языка Python. Перед применением каждого метода очень важно внимательно прочитать документацию для каждого метода, иначе в результате вашего изучения выйдет мало практической пользы.

Доступна различная документация: на сайте документации Python 3, для утилиты командной строки `rudos3` и для оболочки Python 3. В большинстве случаев на сайте документации Python 3 приведена самая свежая информация. Важно попытаться понять представленную информацию и тонкости

методов, которые вы хотите использовать. Иногда мелкое изменение превращает рабочую программу в нерабочую.

Если вы хотите подробнее узнать о методе `os.scandir()`, зайдите на сайт <https://docs.python.org/3/library/os.html#os.scandir>.

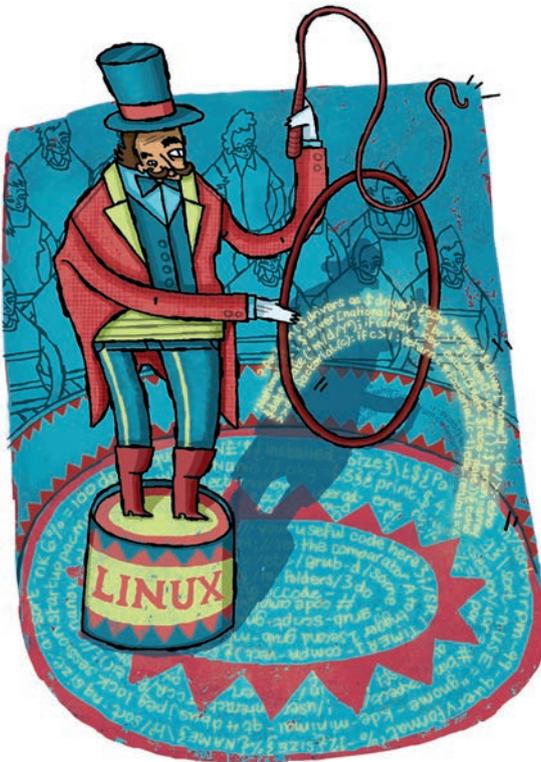
D3.js: Всякие графики

Михалис Цукалос показывает, как создавать впечатляющие графики с помощью библиотеки JavaScript D3.js.



Наш эксперт

Михалис Цукалос (@mactsouk) имеет степень магистра в информационных технологиях от Университетского колледжа Лондона и степень бакалавра по математике. Он администрирует базы данных, пишет программы и работает в Unix как сущая математическая машина. Вы можете связаться с ним через сайт www.mtsoukalos.eu.



D3.js (Data Driven Documents — документы, управляемые данными) — библиотека JavaScript, с помощью которой можно привязать случайные данные к объектной модели документа (DOM) и применить определяемые данными преобразования к документу.

Наш урок содержит больше кода, чем обычно. Это вызвано тем, что в D3.js требуется много кода для построения графиков, так как для получения графиков требуемого вида необходимо определять преобразования, стили и атрибуты. К счастью, этот код можно использовать повторно.

D3.js можно загрузить так:

```
$ wget https://github.com/d3/d3/releases/download/v4.3.0/d3.zip
...
Downloaded: 1 files, 228K in 0.6s (357 KB/s)
$ ls -l d3.zip
-rw-r--r-- 1 mtsouk mtsouk 233712 Oct 27 21:45 d3.zip
$ unzip d3.zip -d d3
...
$ cd d3
$ ls -l d3.min.js d3.js
-rw-r--r-- 1 mtsouk mtsouk 449721 Oct 27 11:43 d3.js
-rw-r--r-- 1 mtsouk mtsouk 213715 Oct 27 11:43 d3.min.js
```

Распаковав файл **d3.zip**, вы получите несколько файлов, в том числе **d3.js** и **d3.min.js**. Эти два файла содержат совершенно одинаковый код JavaScript. Однако файл **d3.min.js** меньше **d3.js** и поэтому загружается быстрее, так как не содержит пробелов.

Чтобы воспользоваться D3, надо включить в HTML-файл одну из двух следующих строк:

```
<script type="text/javascript" src="d3/d3.min.js"></script>
<script src="https://d3js.org/d3.v4.min.js"></script>
```

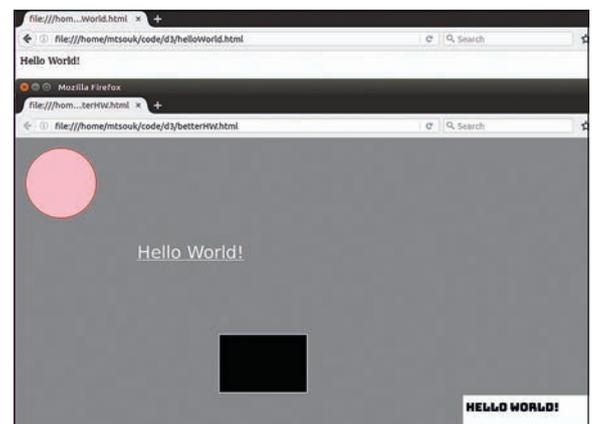
В первом случае используется локальная копия **D3.js**, во втором — копия **D3.js** загружается из Интернета. Преимущество первого метода заключается в том, что вы можете протестировать свой код на локальном компьютере без использования web-сервера или интернет-соединения; зато во втором случае вы всегда получите свежую версию **D3.js**. В первой строке файла **d3.min.js** указан номер версии библиотеки:

```
$ head -1 d3.min.js
// https://d3js.org Version 4.3.0. Copyright 2016 Mike Bostock.
```

Дополнительную информацию о D3.js можно найти на сайте <https://d3js.org>. Чтобы получить побольше вдохновения, зайдите на сайт <https://github.com/d3/d3/wiki/Gallery>.

В следующем коде, который сохранен в файле **helloWorld.html**, представлена версия программы "Hello World", выполненная с помощью D3.js:

```
<!DOCTYPE html>
<meta charset="utf-8">
<body>
  <script type="text/javascript" src="d3/d3.min.js"></script>
<script>
  d3.select("body").append("span")
    .text("Hello World!");
</script>
```



➤ На рисунке показан вывод от **helloWorld.html**, **betterHW.html** и **helloWorldFont.html**.

Скорая помощь

Хотя уже вышла четвертая версия D3.js, предыдущая версия (третья) тоже используется. Между двумя версиями есть некоторые несовместимости, они описаны на странице http://bit.ly/D3_D4Changes. По возможности используйте последнюю версию.

Ряд удобных функций D3.js

Структура D3.js требует писать больше кода, но и обеспечит большую гибкость, особенно если вы создаете какой-нибудь экзотический проект.

Функция `d3.timeParse()` определяет строку анализа-парсинга, которая будет использоваться для преобразования строки в фактическую дату, а для чтения данных из массива `anArray` используется функция `.data(anArray)`.

Функция `d3.scaleLinear()` выполняет линейное масштабирование, принимая в качестве входных

данных область (диапазон значений) и возвращая в качестве выходных данных другой диапазон. Другие функции масштабирования работают похожим образом, но в каждой из них используется свой выходной диапазон, определяемый функцией `range()`.

Функция `d3.axisBottom(x)` рисует оси X с использованием значений, определенных в переменной `x`. Аналогично, `d3.axisLeft(y)` рисует оси Y с использованием значений переменной `y`. Помните, что данные отображаются сверху вниз (при этом

изменяется способ использования функции `range()` для переменной `y`) и что по умолчанию используется черный цвет. Поскольку для осей и обозначений тоже нужно место на экране, потребуются определить дополнительные переменные, определяющие границы ваших выходных данных.

При изменении координат и масштабирования соблюдайте осторожность, так как неверное значение может исказить вид результата до нераспознаваемости.

Простой пример

HTML-код служит оберткой JavaScript, а для вывода сообщения используется объект JavaScript, D3. При всей простоте примера `helloWorld.html`, возможности D3.js гораздо шире. Следующий код представляет собой улучшенную версию программы "Hello World", сохраненную в файле `betterHW.html`:

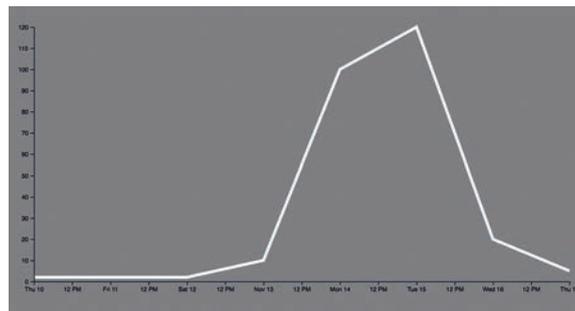
```

<!DOCTYPE html>
<meta charset="utf-8">
<body>
<script type="text/javascript" src="d3/d3.min.js"></script>
<style class="cp-pen-styles">body {
  background-color: #888;
}
</style>
</body>
</html>
<script>
var svgContainer = d3.select("body").append("svg")
  .attr("width", 600).attr("height", 500);
svgContainer.append("svg:rect")
  .style("stroke", "white")
  .attr("x", 340)
  .attr("y", 330)
  .attr("width", 150)
  .attr("height", 100);
svgContainer.append("svg:circle")
  .style("stroke", "red")
  .style("fill", "pink")
  .attr("r", 60)
  .attr("cx", 70)
  .attr("cy", 70);
svgContainer.append("svg:text")
  .style("font-family", "sans-serif")
  .style("text-decoration", "underline")
  .style("font-size", "30px")
  .style("fill", "white")
  .attr("x", 200)
  .attr("y", 200)
  .text("Hello World!");
</script>

```

(Вывод от `helloWorld.html` и `betterHW.html` см. на рис. внизу стр. 92). В файле `betterHW.html` на экране заодно создаются окружность и прямоугольник, и страница становится поинтереснее. [Ред.: — Кхе...]

Прежде всего, в файле `betterHW.html` определяется контейнер, который будет содержать весь вывод. Для добавления новых элементов в существующий контейнер используется функция `append()`. Окружность рисует команда `svg:circle`, а команда



➤ На примере `scale.html` вы видите, как изменить масштаб вывода, чтобы он уместился в контуре.

`svg:rect` — прямоугольник. Чтобы нарисовать окружность, надо задать координаты центра окружности (`cx` и `cy`), а также длину радиуса (`r`). Точно так же, чтобы создать прямоугольник, надо определить его ширину и высоту (`width` и `height`), а также координаты левого верхнего угла (`x` и `y`). Текст добавляется командой `svg:text`, для нее тоже надо указать координаты начальной точки (`x` и `y`). Учтите, что в D3.js не определен шрифт по умолчанию — он задается браузером операционной системы. Для дополнительных функций нужен и дополнительный код, поэтому файл `betterHW.html` намного больше `helloWorld.html`.

Масштабирование данных

Обычно вы не знаете количества элементов для размещения на графике или минимальные и максимальные значения этих элементов. А если вы попытаетесь нарисовать нечто выходящее за пределы контура, оно будет невидимо. Чтобы избежать подобных ситуаций, надо масштабировать данные в соответствии с ограничением конечного экрана. Код файла `scale.html` иллюстрирует этот метод:

```

<!DOCTYPE html>
<meta charset="utf-8">
<style> scalePow
  .line {
    fill: none;
    stroke: white;
    stroke-width: 5px; }
</style>
<body>
<script type="text/javascript" src="d3/d3.min.js"></script>
<script>
var margin = {top: 30, right: 30, bottom: 50, left: 50},
    width = 1024 - margin.left - margin.right,
    height = 768 - margin.top - margin.bottom;
var parseTime = d3.timeParse("%d/%m/%Y");

```

Скорая помощь

SVG — сокращение от «масштабируемая векторная графика [Scalable Vector Graphics]». Это векторный формат изображений, используемый для двумерной графики с поддержкой интерактивности и анимации на основе XML. Дополнительные сведения об SVG см. на <http://www.w3.org/TR/SVG>.

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



Если вы не знакомы с JavaScript и моделью DOM, код D3.js может оказаться трудным для понимания. Хотя для полного понимания, возможно, придется вникнуть в модель DOM и JavaScript, ничто не мешает вам экспериментировать с кодом нашего урока для изучения D3.js.

```
// Определим масштабы
var x = d3.scaleTime().range([0, width]);
var y = d3.scaleLinear().range([height, 0]);
var valueline = d3.line()
  .x(function(d) { return x(d.date); })
  .y(function(d) { return y(d.value); });
var svg = d3.select("body").append("svg")
  .attr("width", width + margin.left + margin.right)
  .attr("height", height + margin.top + margin.bottom)
  .style("background-color", 'grey')
  .append("g")
  .attr("transform", "translate(" + margin.left + "," + margin.top + ")");
// Определим данные
var data = [ {value: 2, date: "10/11/2016"},
  {value: 2, date: "11/11/2016"}, {value: 2, date: "12/11/2016"},
  {value: 10, date: "13/11/2016"}, {value: 100, date: "14/11/2016"},
  {value: 120, date: "15/11/2016"}, {value: 20, date: "16/11/2016"},
  {value: 5, date: "17/11/2016"} ];
// Сделаем строку даты действительно датой!
data.forEach(function(d) {
  d.date = parseTime(d.date);
});
// Масштабируем диапазон данных
x.domain(d3.extent(data, function(d) { return d.date; }));
y.domain([0, d3.max(data, function(d) { return d.value; })]);
svg.append("path")
  .data([data])
  .attr("class", "line")
  .attr("d", valueline);
// Ось X
svg.append("g")
  .attr("transform", "translate(0," + height + ")")
  .call(d3.axisBottom(x));
// Ось Y
svg.append("g")
  .call(d3.axisLeft(y));
</script>
</body>
```

При рисовании линейных данных следует использовать функцию `d3.scaleLinear()`. К другим похожим функциям относятся `d3.scalePow()`, `d3.scaleSqrt()`, `d3.scaleLog()` и `d3.scaleTime()`, которые следует применять для работы с датами и временем. (Вывод `scale.html` см. на стр. 93.) Будь в массиве с данными побольше

элементов, вывод был бы более гладким. Масштабирование применяется в большинстве графиков D3.js, поэтому разберитесь с ним хорошенько.

А теперь посмотрим, как считать данные из внешних текстовых файлов. Это позволит использовать динамические данные. Код файла `readFile.html` таков:

```
<!DOCTYPE html>
<meta charset="utf-8">
<body>
  <script type="text/javascript" src="d3/d3.min.js"></script>
</body>
<script type="text/javascript">
  var data = [];
  var parseTime = d3.timeParse("%Y%m%d%H%M");
  d3.csv("./data.txt", function(error, data) {
    if (error) throw error;
    data.forEach(function(d) {
      d.connections = +d.connections;
      tempDate = "2016"+d.month+d.day+d.hour+d.minute;
      d.date = parseTime(tempDate);
      document.write("<li>"+ d.date + ": " + d.connections);
    });
    console.log(data);
  });
</script>
```

Содержимое файла `data.txt` имеет следующий формат:

```
$ head data.txt
connections,month,day,hour,minute
0,11,20,12,58
2,11,20,13,00
$ wc data.txt
1001 1001 14034 data.txt
```

В первом столбце отображается количество подключений TCP/IP, во втором — месяц, в третьем — день месяца, в четвертом — час, и в последнем — минута. Так как файл `data.txt` представляет собой файл CSV, для чтения этого файла мы воспользуемся функцией `d3.csv()`. Функция `forEach()` используется для перебора всех элементов массива данных и для преобразования каждого строкового значения в целое число (`d.connections = +d.connections`), поскольку строки на график не нанесешь. Кроме того, вам потребуется объединить четыре последних значения каждой строки, проанализировать их и преобразовать в дату. Так как в файле `data.txt` нет столбца с годом, год придется ввести вручную.

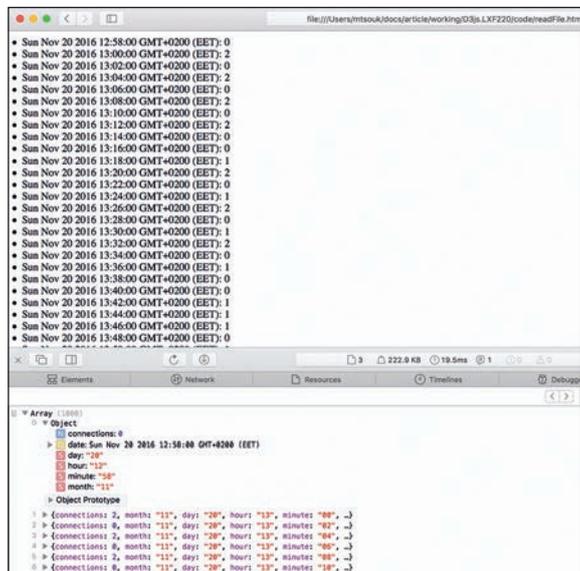
Хотя чтение внешних файлов — полезный навык, для экономии места мы будем использовать только статические данные. Тем не менее, в большинстве случаев вы будете получать данные из Интернета или из текстового файла, поскольку в рисовании статических данных нет никакой гибкости. (На рис. слева показаны результаты для файла `readFile.html`, а также вывод консоли JavaScript.) Теперь, когда вы снова захотите воспользоваться файлом `readFile.html`, понадобится только изменить данные файла `data.txt`, не меняя исходный файл. Это основное преимущество данного подхода.

Круговые диаграммы

С помощью D3.js можно создавать круговые диаграммы, и файл `pieChart.html` научит вас, как это делается. В файле `pieChart.html` важен следующий код:

```
var colors = ["#98abc5", "#445544", "#bbaadd", "#aaff55"];
var arc = d3.arc()
  .outerRadius(radius - 10)
  .innerRadius(0)
  .context(context);
var labelArc = d3.arc()
  .outerRadius(radius - 100)
  .innerRadius(radius - 100)
```

➤ **Результат для readFile.html, из которого вы узнали, как с помощью D3.js считывать текстовые файлы. Откройте в своем web-браузере консоль JavaScript, в ней очень удобно разрабатывать приложения D3.js.**



```

.context(context);
var pie = d3.pie()
.sort(null)
.value(function(d) { return d.value; });
context.translate(width / 2, height / 2);
var data = [ {title: "Mihalis", value: 100},
{title: "Dimitris", value: 200},
{title: "John", value: 140},
{title: "Maria", value: 120},
{title: "Helen", value: 211},
{title: "Neil", value: 123},
{title: "April", value: 123}
];
var arcs = pie(data);
arcs.forEach(function(d, i) {
context.beginPath();
arc(d);
var temp = i % colors.length;
context.fillStyle = colors[temp];
context.fill();
});
context.beginPath();
arcs.forEach(arc);
context.strokeStyle = "#000";
context.stroke();
context.textAlign = "center";
context.textBaseline = "middle";
context.fillStyle = "#000";
arcs.forEach(function(d) {
var c = labelArc.centroid(d);
context.fillText(d.data.title, c[0], c[1]);
});

```

Массив `colors` содержит используемые цвета. В массив данных можно добавить сколько угодно записей, при условии, что на экране будет достаточно места для вывода всех сегментов круговой диаграммы без перекрывания. (Созданную круговую диаграмму см. на рис. вверху.)

Построение гистограммы

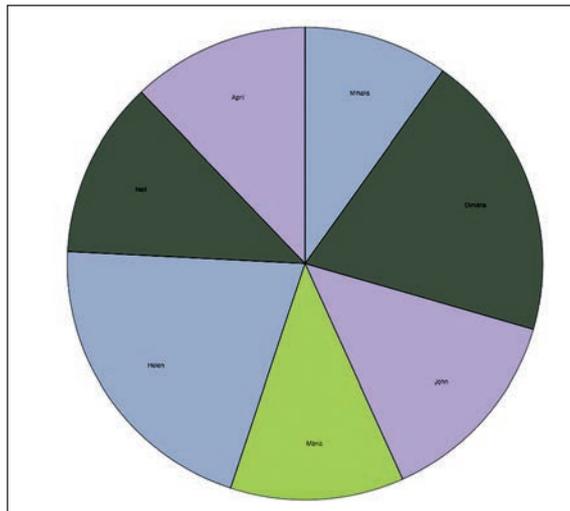
Построение по вашим данным гистограммы (столбчатой диаграммы) — очень распространенный способ представления данных. Построение гистограммы иллюстрируется в коде `barChart.html`. Основной код в этом файле таков:

```

svg.selectAll(".bar")
.data(data)
.enter().append("rect")
.attr("class", "bar")
.attr("x", function(d) { return x(d.title); })
.attr("width", x.bandwidth())
.attr("y", function(d) { return y(d.value); })
.attr("height", function(d) { return height - y(d.value); });

```

Атрибут `x` используется для заголовка столбца, `y` задает значение столбца; каждый столбец представляет собой прямоугольник.



► Результат для `pieChart.html`, из которого вы узнаете, как строить с помощью D3.js круговые диаграммы.

Опять же, данные надо масштабировать, чтобы они уместились в контуре.

Покажем, как преобразовать вертикальную гистограмму в горизонтальную. Вместо того, чтобы приводить весь код файла `horizontalBC.html`, продемонстрируем различия между файлами `horizontalBC.html` и `barChart.html`, сформированные утилитой `diff`:

```

$ diff barChart.html horizontalBC.html
16c16
< var margin = {top: 20, right: 20, bottom: 30, left: 40},
---
> var margin = {top: 20, right: 20, bottom: 30, left: 60},
20,21c20,21
< var x = d3.scaleBand().range([0, width]).padding(0.1);
< var y = d3.scaleLinear().range([height, 0]);
---
> var x = d3.scaleLinear().range([0, width]);
> var y = d3.scaleBand().range([height, 0]).padding(0.1);
33,34c33,34
< .attr("width", width + margin.left + margin.right)
< .attr("height", height + margin.top + margin.bottom)
---
> .attr("width", W - margin.left - margin.right)
> .attr("height", H - margin.top - margin.bottom)
38,39c38,39
< x.domain(data.map(function(d) { return d.title; }));
< y.domain([0, d3.max(data, function(d) { return d.value; })]);
...

```

Как видите, надо немного изменить левое поле, чтобы вместились больше текста. Так как необходимо разместить обозначения на оси Y и значения на оси X, следует также изменить параметры масштабирования.

Если вы уже знакомы с JavaScript, то легко воспользуетесь D3.js. Основное преимущество D3.js по сравнению с программами вроде R состоит в том, что D3.js подходит для web-разработки. **LXF**

Использование шрифтов Google

Для украшения вывода HTML можно воспользоваться шрифтами Google. (Просмотреть полный список шрифтов и выбрать интересующий вас шрифт можно на сайте <https://fonts.google.com>).

После выбора шрифта Google сообщит, какой HTML-код надо включить в файлы для использования этого шрифта. После включения необходимого

кода потребуется добавить еще немного кода для использования шрифта.

Например, если вы выбрали шрифт Bungee Inline, сначала надо поместить следующий код в раздел `<head>` HTML-документа, чтобы загрузить шрифт:

```

<link href="https://fonts.googleapis.com/css?family=Bungee+Inline" rel="stylesheet">

```

После этого добавьте следующий код CSS, чтобы изменить шрифт текста:

```

.style("font-family", "Bungee Inline, cursive")

```

Вывод файла `helloWorld.html` со шрифтом Bungee Inline можно увидеть на рисунке в правом нижнем углу стр. 92. Чтобы увидеть полную реализацию, взгляните на HTML-код файла `helloWorldFont.html`.

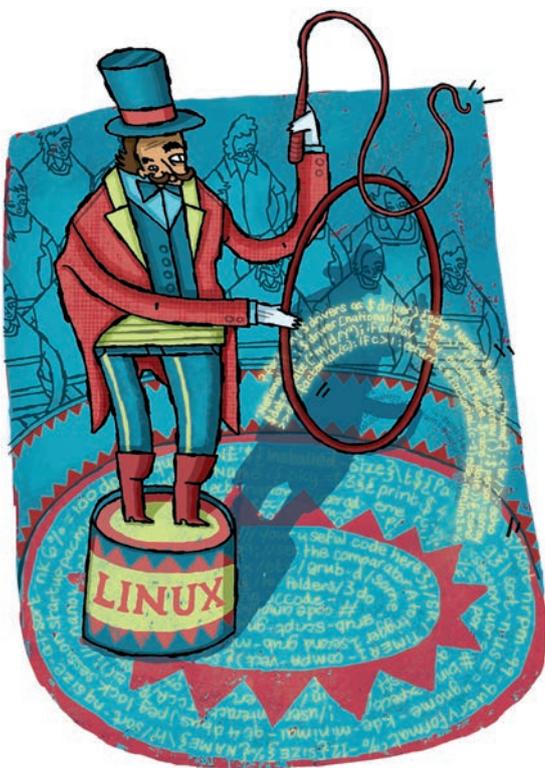
ФРАКТАЛЫ: Нарисуем красоту

Михалис Цукалос учит рисовать некоторые известные фракталы, в том числе ваши собственные снежинки, на разных языках программирования.



Наш эксперт

Михалис Цукалос (@mactsouk) имеет степень магистра в информационных технологиях от Университетского колледжа Лондона и степень бакалавра по математике. Он администрирует базы данных, пишет программы и работает в Unix как сущая математическая машина.



произвести огромные изменения в его функции. Функция — это то, что нам здесь важно, потому что способ, которым вы чертите функцию, как раз и производит сам фрактал. Динамические системы очень чувствительны к начальным условиям, это и есть причина возникновения хаоса. Это означает, что предсказать их поведение невозможно, даже если они ведут себя детерминированно, потому что их поведение полностью зависит от начальных условий. Другими словами, хотя настоящее и определяет будущее, приблизительное настоящее этого сделать не сможет. Теория хаоса применяется в различных областях, включая компьютерную безопасность, метеорологию, инженерию, биологию и информатику, а следующий раздел представит вам различные хорошо известные фракталы.

Термин фрактал впервые (в 1975 г.) ввел математик Бенуа Мандельброт [Benoit Mandelbrot]; итак, не удивительно, что первый и самый знаменитый фрактал был назван в честь него — фрактал Мандельброта. Второй по известности — фрактал Жюлиа [Julia], а за ним следует снежинка Коха [Koch Snowflake]. Фрактал Коха начинается с отрезка прямой линии. На нашем уроке мы также представим вам треугольник Серпинского [Sierpinski Triangle].

У каждого фрактала есть параметры, которые для его получения необходимо задать. Если вы используете для создания фрактала модуль, вам надо лишь определить эти параметры, и всё

Скорая помощь

Две продвинутые книги, которые научат вас основной части теории динамических систем и хаоса — «Введение в прикладные нелинейные динамические системы и хаос [Introduction to Applied Nonlinear Dynamical Systems and Chaos]», автор Стивен Уиггинс [Stephen Wiggins], и «Дифференциальные уравнения и динамические системы [Differential Equations and Dynamical Systems]», автор Лоуренс Перко [Lawrence Perko].

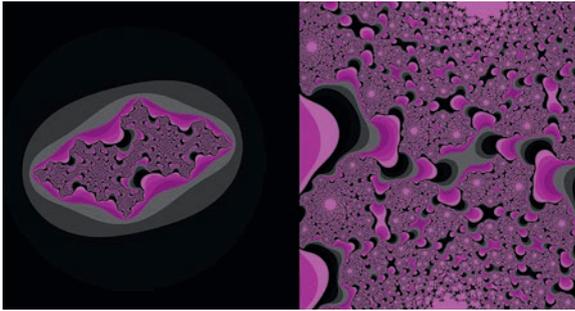
Тема этой статьи — нечто живописное и впечатляющее: фракталы. Разобравшись в необходимой теории, вы научитесь моделировать невероятно красивые изображения, используя различные языки программирования. Мы рассмотрим скрипты на языках R, Perl, Python и LaTeX, так что вы легко сможете перенести примеры на выбранный вами язык программирования. Хотя фракталы и теория хаоса — области высшей математики, временами они могут приносить и удовольствие, порождая чудесные картинки — даже если вы ничего не поймете из представленной теории, они должны вам понравиться!

Фрактал — это природное явление и математическое множество. Фрактал представляет собой повторяющийся узор, воспроизводимый в любом масштабе. Это означает, что точно такой же узор может быть обнаружен, даже если масштаб многократно увеличить, что делает фракталы уникальными и приятными для глаз. Большинство фракталов начинаются с математического уравнения, которое, будучи зарисованным некоторым способом, создает интересный, но довольно сложный график.

Фракталы относятся к областям высшей математики, занятым динамическими системами и хаосом. По-простому, теория хаоса говорит нам, что малые изменения в заданном количестве могут



► Perl-скрипт mandelSet.pl демонстрирует простой способ рисования фрактала Мандельброта через текстовый вывод.



» Здесь показан вывод от двух выполнений скрипта `juliaSet.py` для разных значений переменных `xa`, `xb`, `ya` и `yb`. Фрактал справа — увеличенная версия левого.

пойдет само. С другой стороны, если вы предпочитаете делать работу лично, то можете всё определить, тогда у вас будет больше возможностей для контроля и лучшего понимания процесса, хотя потребуются больше кода. Большинство скриптов в этой статье модуль не используют.

Мандельброт изнутри

Множество Мандельброта — один из самых известных фракталов. Давайте кратко поясним, как это работает. Используемая функция называется квадратный многочлен. Формула квадратного многочлена — $f(x) = x^2 + c$, где c — постоянное число. Чтобы начать генерировать последовательность, понадобится стартовое число, называемое x_0 , которое называют «затравкой [seed]» — начальным значением итерации. После вычисления значения $f(x_0)$, которое обозначается x_1 и равняется $x_0^2 + c$, вычисляется следующее — $f(x_1)$, оно обозначается x_2 и равняется $x_1^2 + c$. Ну и так далее. Представьте теперь, что c является комплексным числом, то есть числом, представимым как $a + bi$, где a и b — вещественные числа, а i — мнимая единица. Все показанные здесь алгоритмы, которые рисуют фрактал Мандельброта, образуются заменой этих двух вещественных чисел (a и b). (Если кто забыл, мнимая единица i удовлетворяет уравнению $i^2 = -1$.) Итак, чтобы породить последовательность Мандельброта, следует выбрать начальное значение и постоянную комплексную величину c .

Строго говоря, множество Мандельброта состоит из всех (комплексных) значений c , для которых последовательность от квадратного многочлена $x^2 + c$ не стремится к бесконечности. Фраза «не стремится к бесконечности» означает, что значение $x^2 + c$ не возрастает при каждой итерации, а убывает. Вот так примерно, и относительно просто, работает множество Мандельброта. Это звучит сухой теорией, пока вы не увидите чарующие графики, которые оно производит! Множество Жюлиа основано на множестве Мандельброта, и основная идея в нем та же.

Реализация на Perl

Perl может помочь вам понадевать собственных фракталов. Ради упрощения работы представленный скрипт Perl будет вместо графического файла рисовать текстовый вывод! Код Perl `mandelSet.pl` следующий:

```
#!/usr/bin/perl -w
use Math::Complex;
sub MB {
    my ($fx, $c) = @_ [0,0];
    for(1 .. 20) {
        $fx = $fx * $fx + $c;
        return $_ if abs $fx > 2;
    }
}
```

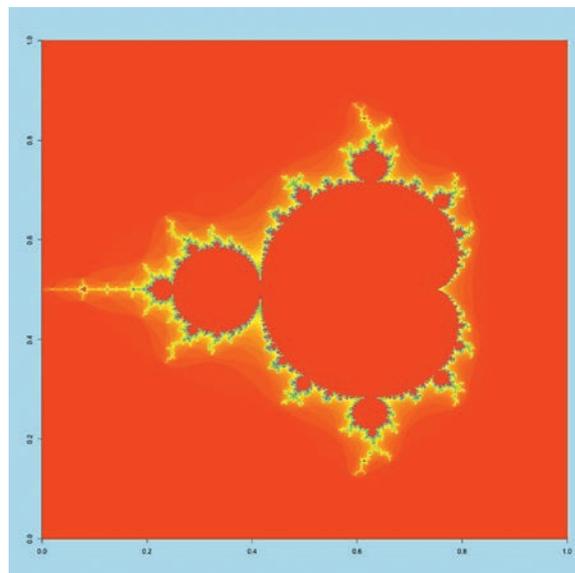
```
}
my $y = 0;
for ($y = 1; $y >= -1; $y -= 0.05) {
    my $x = 0;
    for ($x = -2; $x <= 0.5; $x += 0.0305) {
        print MB($x + $y * i) ? ' ': '#';
    }
    print "\n";
}
```

(см. внизу стр. 96 текстовый вывод этого скрипта). Ключевой момент скрипта Perl — это обработка комплексных чисел с помощью модуля `Math::Complex`. Применяемый квадратный многочлен определяется функцией `MB`, использующей переменную `$fx`. Скрипт Perl — простенький способ произвести впечатление на друзей и лучше разобраться в процессе. Приведенный далее код Python добавит выводу красок и сделает его еще эффективнее.

Реализация на Python

В этом разделе мы представим два скрипта Python 3, описывающих множество Жюлиа и множество Мандельброта. Первый скрипт, с именем `juliaSet.py`, рисует множество Жюлиа:

```
#!/usr/bin/env python3
import sys
from PIL import Image
import random
if len(sys.argv) == 2:
    outputSize = int(sys.argv[1])
else:
    print('Usage: ', sys.argv[0], 'ImageSize')
    sys.exit(0)
image = Image.new("RGB", (outputSize, outputSize))
xa = -2.5
xb = 2.5
ya = -2.5
yb = 2.5
nIterations = 128
while True:
    cx = random.random() * (xb - xa) + xa
    cy = random.random() * (yb - ya) + ya
    c = cx + cy * 1j
    z = c
```



Скорая помощь

Узнать больше о множестве Мандельброта и множестве Жюлиа можно здесь: https://en.wikipedia.org/wiki/Mandelbrot_set и https://en.wikipedia.org/wiki/Julia_set, соответственно. А о снежинке Коха — здесь: https://en.wikipedia.org/wiki/Koch_snowflake.

»

» Это вывод скрипта `fr.R`, действовавшего для рисования фрактала Мандельброта возможности языка R.

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

Скорая помощь

Основная идея в том, чтобы вы использовали наиболее знакомый язык программирования. Кроме того, когда имеешь дело с большим количеством математики, ПО типа R, *Octave* и *MatLab* может сэкономить кучу времени, потому что все пакеты в свободном доступе имеют встроенную поддержку большинства математических формул и функций.

```

for i in range(nIterations):
    if abs(z) > 2.0:
        break
    z = z * z + c
    if i > 10 and i < 100:
        break
for y in range(outputSize):
    zy = y * (yb - ya) / (outputSize - 1) + ya
    for x in range(outputSize):
        zx = x * (xb - xa) / (outputSize - 1) + xa
        z = zx + zy * 1j
        for i in range(nIterations):
            if abs(z) > 2.0:
                break
            z = z * z + c
        image.putpixel((x, y), (i % 8 * 32, i % 4 * 32, i % 8 * 32))
image.save("juliaPython.png", "PNG")

```

Цикл `while` в скрипте **juliaSet.py** использует множество Мандельброта для нахождения точек множества Жюлиа. Вывод был получен при таком запуске скрипта:

```

$ ./juliaSet.py 1600

```

Как можно видеть на рис. стр. 97, вывод содержит два изображения. Правое изображение — это увеличенная версия левого. На правом изображении величины `xb` и `yb` были 0,5, а величины `xa` и `ya` были -0,5. Можно увеличивать сколько угодно, используя величины 0,05 для `xb` и `yb` и -0,05 для `xa` и `ya`. Точно так же можно нарисовать фрактал Мандельброта с помощью скрипта **mandelSet.py**:

```

#!/usr/bin/env python3
from numpy import complex, array
from PIL import Image
import colorsys
import sys
if len(sys.argv) == 2:
    outputSize = int(sys.argv[1])
else:
    print("Usage: ", sys.argv[0], 'ImageSize')
    sys.exit(0)
def iToRGB(i):
    color = 255 * array(colorsys.hsv_to_rgb(i/255.0, i/100.0, i/55.0))
    return tuple(color.astype(int))
def M(x, y):
    c0 = complex(x, y)
    c = 0
    for i in range(1, 1000):
        if abs(c) > 2:
            return iToRGB(i)
        c = c * c + c0
    return (10, 10, 10)
img = Image.new('RGB', (outputSize, int(outputSize/2)))
pixels = img.load()
for x in range(img.size[0]):
    for y in range(img.size[1]):
        pixels[x,y] = M((x - (0.75 * outputSize))/(outputSize/4),
            (y - (outputSize/4))/(outputSize/4))

```

```

img.save('mandelPython.png', 'PNG')

```

Квадратный многочлен внутри функции `M()` определен с помощью переменной `c`. Функция принимает два параметра — вещественных числа, это действительная и мнимая части комплексной константы в квадратном многочлене. Переменная `c0` хранит значение комплексной константы. Скрипт **mandelSet.py** принимает один параметр — ширину конечного изображения в пикселях, и сохраняет конечное изображение как **mandelPython.png**:

```

$ ./mandelSet.py 1600

```

У R есть большое преимущество перед всеми программными языками: он знаком с математикой [Ред.: — Как с Кун-фу?], а значит, можно определить какую угодно функцию и довольно легко нарисовать ее любым понравившимся способом. Взгляните на следующий R-скрипт, с именем **fr.R**:

```

#!/usr/bin/env Rscript
MB <- function(z, c, trans, cond, max=50, response=dwell)
{
    active <- seq_along(z)
    dwell <- z
    dwell[] <- 0
    for (i in 1:max)
    {
        z[active] <- trans(z[active], c[active]);
        survived <- cond(z[active])
        dwell[active[!survived]] <- i
        active <- active[survived]
        if (length(active) == 0) break
    }
    eval(substitute(response))
}
re = seq(-2, 1, len=1000)
im = seq(-1.5, 1.5, len=1000)
c <- outer(re, im, function(x,y) complex(real=x, imaginary=y))
x <- MB(array(0, dim(c)), c, function(z,c)z^2+c, function(z) abs(z)
<= 2, max=50)
png(filename="Rfractal.png", width=1024, height=1024)
par(bg = "lightblue")
par(col = "black")
image(x, col=rainbow(100))
dev.off()

```

Скрипт работает прямо с функциями от комплексных чисел — это удобно и спасает от написания дополнительного кода или подключения дополнительного модуля. Третий параметр переменной `x` — это определение функции квадратного многочлена, а переменная `c` хранит различные величины комплексной константы. На рис. внизу стр. 97 показан вывод R-кода. Этот красочный вывод производит функция `rainbow()` внутри функции `image()`. Если вы хотите ограничиться в выводе всего двумя цветами, вызов функции `rainbow()` можно удалить.

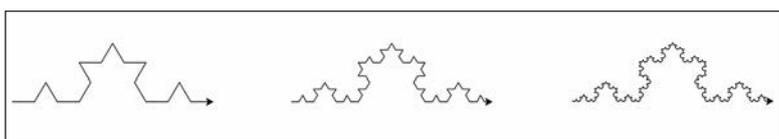
Рисуем снежинку Коха

Снежинка Коха начинается с отрезка прямой, т.е. это ближе к геометрии, чем к алгебре. Каждая итерация заменяет среднюю треть каждого отрезка парой других отрезков, образуя равносторонний треугольник, который направлен вовне и лишен основания. Теперь, разобравшись, как устроена снежинка Коха, пора просмотреть код Python 3 скрипта **kochTr.py**:

```

#!/usr/bin/env python3
import os
import sys
from turtle import *
def snowflake(length, levels):

```



➤ Эти три вывода от скрипта **kochTr.py** на Python 3 должны дать лучшее понимание построения снежинки Коха.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

```

if levels == 0:
    fd(length)
    return
length /= 3.0
snowflake(length, levels-1)
left(60)
snowflake(length, levels-1)
right(120)
snowflake(length, levels-1)
left(60)
snowflake(length, levels-1)
if len(sys.argv) == 3:
    length = int(sys.argv[1])
    levels = int(sys.argv[2])
penup()
bk(length/2.0)
pendown()
snowflake(length, levels)
mainloop()
else:
    print('Usage: ', sys.argv[0], 'Length Levels')
    sys.exit(0)

```

Для начертания конечного изображения скрипт использует черепашку графики. На рис. внизу стр. 98 приведены три отдельных графика, для лучшего понимания динамики стадий этого процесса:

```

$ ./kochTr.py 300 2
$ ./kochTr.py 300 3
$ ./kochTr.py 300 4

```

Поразительно, как столь простая идея создает такой потрясающий график!

О, треугольник Серпинского

Следующий скрипт Python 3, `sierpinski.py`, рисует треугольник Серпинского:

```

#!/usr/bin/env python3
import turtle
import os
import sys
def S(side, level):
    turtle.bgcolor("lightgreen")
    if level == 1:
        for i in range(3):
            turtle.fd(side)
            turtle.left(120)
    else:
        S(side/2, level-1)
        turtle.fd(side/2)
        S(side/2, level-1)
        turtle.bk(side/2)
        turtle.left(60)
        turtle.fd(side/2)
        turtle.right(60)
        S(side/2, level-1)
        turtle.left(60)
        turtle.bk(side/2)
        turtle.right(60)
if len(sys.argv) == 3:
    s = int(sys.argv[1])
    l = int(sys.argv[2])
    S(s, l)
    turtle.mainloop()
else:
    print('Usage: ', sys.argv[0], 'Side Level')
    sys.exit(0)

```

О треугольнике Серпинского

Треугольник Серпинского — еще один изобретательный способ сделать фрактал, который основан на геометрии и работает с равносторонними треугольниками, рекурсивно подразделяемыми на меньшие равносторонние треугольники.

Алгоритм прост для понимания и работает следующим образом: взяв равносторонний треугольник, вы делите его на четыре равных меньших равносторонних треугольника. Затем удаляете

центральный треугольник — он остается пустым и всегда видимым, потому что в процессе больше не участвует. Наконец, делаете то же самое с оставшимися тремя равносторонними треугольниками столько раз, сколько нужно.

Конечный продукт фрактала «треугольник Серпинского» всё еще состоит из многочисленных равносторонних треугольников. На рис. внизу страницы это показано в действии.

Первый параметр функции `S()` определяет размер конечного треугольника, а второй параметр — уровень рекурсии треугольника. См. внизу вывод предыдущего скрипта, когда он выполнен дважды следующим образом:

```

$ time ./sierpinski.py 500 5
real    1m46.462s
user    0m12.508s
sys     0m3.427s
$ time ./sierpinski.py 500 7
real    14m43.285s
user    1m25.576s
sys     0m21.511s

```

Также можно увидеть вывод команды `time` для каждого запуска. Фактическое время не имеет значения — важно то, что первое выполнение значительно быстрее (примерно в восемь раз), чем второе, поскольку второе выполнение усложнено. Тем не менее, размер внешнего треугольника в обоих случаях одинаковый, потому что первый параметр тот же. Причем вследствие способа работы черепашкой графики процесс анимирован и потому выглядит еще более впечатляющим, чем сам результат скрипта Python.

В завершение скажу, что для LaTeX есть некоторые готовые к употреблению пакеты, умеющие рисовать фракталы, включая фрактал Мандельброта:

```

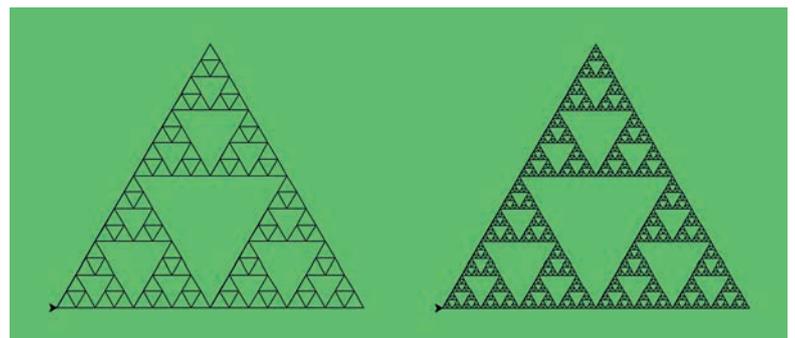
\documentclass{article}
\usepackage{tikz}
\usetikzlibrary{shadings}
\begin{document}
\tikz\shade[shading=Mandelbrot set] (0,0) rectangle (15,15);
\end{document}

```

Чтобы увидеть сгенерированный вывод, надо будет скомпилировать код LaTeX, сохранив его как `mandelbrot.tex`:

```
$ pdflatex mandelbrot.tex
```

И выводом скрипта будет `mandelbrot.pdf`. Как видите, фракталы — это весело!



➤ Здесь представлены два различных по степени сложности треугольника Серпинского. Однако внешние треугольники одинакового размера.



LXF HotPicks



Александр Толстой

Проводит время, ныряя с коралловых рифов за жемчужинами для HotPicks. Если только не делает вид, что взламывает результаты выборов.

AzPainter » KSnip » QRab » Findimagedupes » Cloudcmd » Latte Dock » Argos » Machines vs Machines » 404 Elm Street » GrSync » Buttercup

Редактор изображений

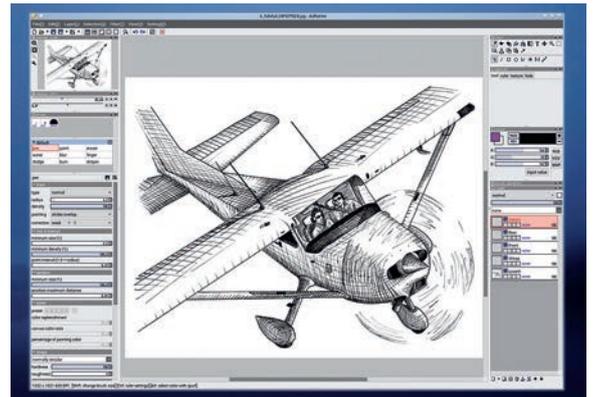
AzPainter

Версия: 2.0.2 Сайт: <http://bit.ly/2m2wQPN>

Герой данного обзора — крошечная программка, исходный код которой занимает всего половину старомодного 3,5" флоппи-диска, или 750 КБ. Но при этом он конкурирует с основными редакторами изображений, такими, как GIMP или Krita — признанными приложениями-тяжеловесами Linux для энтузиастов графики и художников.

AzPainter был написан японским инженером-программистом, известным как Azel, для создания растровых иллюстраций (BMP). Он реализует оригинальную структуру интерфейса пользователя, оптимизированного для эффективного рисования — так, как его видит автор, вот почему

у вас может уйти несколько лишних минут на привыкание к интерфейсу AzPainter. В приложении поддерживаются изображения 16-битным цветом и прозрачностью (RGBA), автоматическое уточнение размера кисти в зависимости от давления ручки планшета, функции слоев в стиле PhotoShop, десятки художественных фильтров, разные типы инструментов выделения, поддержка многих сторонних



» AzPainter — идеальный инструмент для рисования эскизов и создания технических иллюстраций.

Эта программа помещает все функции в крошечный пакет.

форматов изображений, включая PSD, и многое другое.

При всём при том, AzPainter не так потрясал бы, будь он всего лишь очередным редактором изображений для Linux; но эта программа помещает все свои функции в крошечный пакет, способный также похвастаться невероятной скоростью во всём, от времени запуска до общей отзывчивости. Нам кажется, что вряд ли на свете найдется что-нибудь, более противоположное термину «раздутое» ПО, чем AzPainter.

Приложение использует оригинальный и супер-легковесный графический набор инструментов под названием Mlib. Он выглядит очень компактным, в серых тонах, и, естественно, ненавязчив, чтобы вы могли сосредоточиться на контенте. AzPainter очень силен в рисовании сложных фигур и живописи; в нем имеются многочисленные настройки кистей (давление, режим смешивания, интервал, внесение случайности и т.д.), поддержка текстур и множество дополнительных прелестей, как только вы разберетесь с интерфейсом (слева вы найдете ряд подсказок).

В Интернете очень мало бинарников AzPainter, и официальная страница AzPainter — на японском, что несколько усложняет задачу запуска этого приложения в вашей системе. Однако его компиляция из исходника не потребует много зависимостей, и процесс идет довольно быстро с помощью старой доброй последовательности `./configure && make && sudo make install`.

Исследуем интерфейс AzPainter

Вид холста

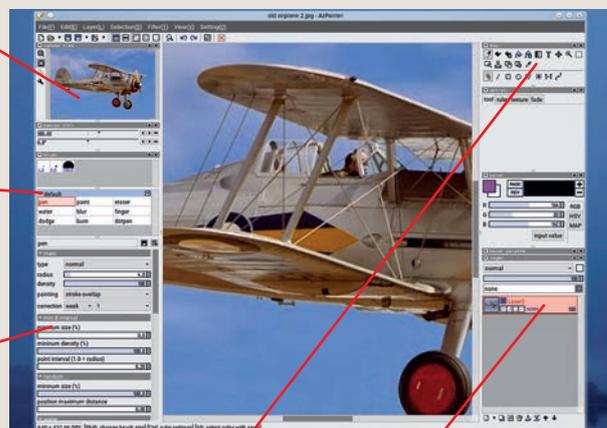
Щелчок здесь поможет вам перемещаться по изображению с измененным масштабом.

Настройка кисти

Выберите тип кисти и уточните ее настройки, например, размер, плотность, нахлест и т.д.

Большая точность живописи

Добавляет более тонкий контроль плотности и интервала кисти.



Другие инструменты

Прямоугольники, штамп, инструменты выделения и многое другое с расширенными опциями для каждого инструмента ниже.

Слои!

AzPainter поддерживает «папки» слоев, такие же, как группы в GIMP и PhotoShop.

Ножницы

KSnip

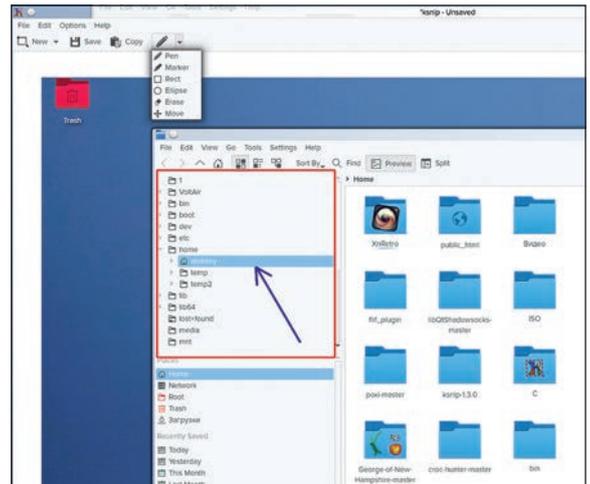
Версия: 1.3 Сайт: <http://bit.ly/2mmadrt>

НotPicks понимает, что почти все пользователи Linux хотят заполнить свой рабочий стол небольшими удобными аксессуарами, которые облегчают повседневную жизнь. На сей раз мы рассмотрим приложение, вдохновителем которого послужили стандартные «ножницы» в Microsoft Windows. Оно дополняет встроенную функцию системы по созданию скриншотов, обычно привязанную к клавише PrintScreen. Хотя вы можете слегка подправить ее поведение (например, Alt+PrtScrn для захвата только активного окна), все остальные полезные функции, например, рисование на снимке стрелок или прямоугольников, доступны только в ножницах или в сторонних коммерческих приложениях.

Примечательно, что рабочим столам Linux по-прежнему не хватает готовых инструментов ножниц или похожих сторонних приложений. Единственным достойным кандидатом является *Shutter* со своим отдельным редактором изображений. Но подождите, у нас есть кое-что получше...

KSnip — это клон ножниц Windows, компактное плавающее окно всегда на переднем плане, которое позволяет делать снимок выбранной части экрана и быстро в ней что-то выделять.

Можно рисовать линии, прямоугольники, эллипсы или использовать неразрушающий инструмент выделения: всё это размещается поверх изображения, и вы всегда можете переместить или удалить любой нарисованный объект. На данный момент там отсутствует инструмент со стрелкой, но можно рисовать прямые линии, удерживая Shift и используя инструмент Pen [Ручка]. В окне настроек *KSnip* вам откроются цвета для рисования по умолчанию и очень полезный инструмент выгрузки в Сеть — Imgur (в нем поддерживается аутентификация).



► **Сделайте ваши снимки экрана более профессиональными с помощью *KSnip*. Мало ли кто их увидит.**

KSnip в данный момент находится в стадии активной разработки. Когда мы впервые на него наткнулись, в нем не было инструментов рисования прямоугольников или эллипсов, но мы связались с автором, и после нашего материала по *KSnip* на странице GitHub тот среагировал очень быстро, реализовав эти функции. По каким-то причинам код придерживается устаревшей версии Qt 4.8 и для успешной компиляции требует GCC 4.8. Разработчик *KSnip* выразил надежду на то, что в среднесрочной перспективе его приложение будет портировано в Qt 5, однако это потребует значительной переработки кода.

Рабочим столам Linux не хватает готовых инструментов-ножниц.

Сканер QR-кода

QRab

Версия: 0.4 Сайт: <https://github.com/SeeLook/QRab>

Программы кодирования QR мы рассматриваем, мягко говоря, не слишком часто. QR означает Quick Response [QR-код — двумерное изображение кодированной информации, позволяющее быстро ее считывать, например, кассовым аппаратом в магазине], а *QRab* — это инструмент-«краб», загребаящий эти коды с вашего экрана. *QRab* задуман как удобный аксессуар, который упростил бы постоянное сканирование QR-кода и, по возможности, рутинную работу, если надо обрабатывать много таких кодов.

Приложение размещается поверх всех окон и в общем не путается у вас под ногами. Если где-то на экране имеется QR-код, вы можете нажать на кнопку Grab, и *QRab* автоматически его декодирует, а текст распечатает и мгновенно скопирует в буфер. Если на экране более одного кода, *QRab* обрабатывает только первый из них (слева сверху). Приложение использует мощный

движок ZBar для работы с разными видами кода, благодаря чему кроме базового QR Code поддерживает EAN-13/UPC-A, UPC-E, EAN-8, Code 128, Code 39, Codabar, Interleaved 2 of 5. Благодаря ZBar *QRab* умеет также распознавать коды, полученные с видеоустройств, так что в качестве сканера штрих-кодов сгодится web-камера.

QRab предлагает очень легкий и простой интерфейс, основанный на управлении Qt Quick, и он очень похож на шустрое мобильное приложение. Вдобавок в нем есть некоторые расширенные опции. Нажмите кнопку Adjust Sheet, чтобы настроить раскладку кодов, которые содержат таблицы — *QRab* позволяет копировать



► **Поиграйте в «кассира супермаркета» со своими детьми!**

и вставлять содержимое QR-кода в электронную таблицу, и вам может понадобиться настроить форматирование кода заранее. Кнопка Settings содержит функцию замены текста. С помощью знака плюс можно добавить строку и отредактировать имеющийся декодированный текст.

Компиляция *QRab* занимает какие-то несколько минут и требует наличия пакетов разработки для Qt Quick Controls и ZBar. Затем процесс идет через такие стандартные команды, как `make` и `make`, и предоставляет вам блестящую новенькую рабочую *QRab*.

QRab умеет распознавать коды, полученные с видеоустройств.

Программа поиска изображений-дубликатов

Findimagedupes

Версия: GIT Сайт: <http://bit.ly/2m6cJRw>

Мы всё время ищем приложения, которые помогли бы оптимизировать огромную коллекцию изображений, собранных за последние годы. Мы уже пробовали многообещающие форматы файлов Lepton и FLIF и наслаждались их превосходным уровнем сжатия (см. LXF215 и LXF205), но, честно говоря, даже идеальное сжатие данных не решает проблемы избыточных файлов.

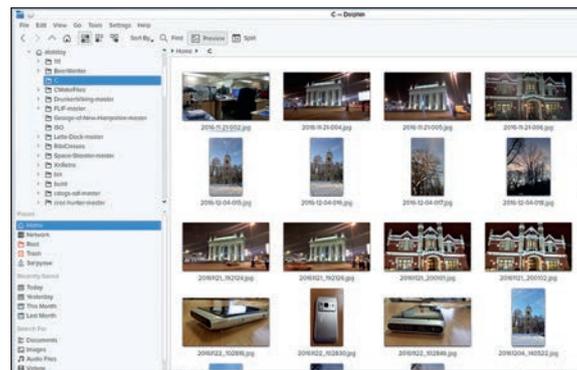
Основная проблема в том, что у нас есть множество фото, и среди них имеются просто дубликаты или же серии практически одинаковых снимков. Таким образом, нам нужно выявлять ненужные файлы и с легкостью от них избавляться. Поиск по репозиторию GitHub выявил множество подобных проектов, основанных на технологии компьютерного распознавания образов, известной как OpenCV, но большинство из них предназначены для Windows и представляют неразрешимые проблемы при попытке скомпилировать код в Linux. К счастью, есть более легкий, но очень эффективный

инструмент под названием *Findimagedupes*. Это небольшая утилита, которая хэширует файлы и может мгновенно определить одинаковые или похожие снимки. Степень сходства задается аргументом утилиты.

Приложение работает в командной строке и написано на Go, а это хорошая новость, поскольку у Go уже есть отличные опции управления пакетами, способные взять код и живенько его скомпилировать. После установки пакета с помощью стандартного менеджера пакетов настройте переменную \$GOPATH:

```
$ export GOPATH=/path/you/choose
Выберите пункт назначения и затем установите Findimagedupes:
$ go get github.com/opennota/Findimagedupes
```

Findimagedupes запоминает все хэши, и работает быстро.



➤ Избавляйтесь от лишних файлов и сохраняйте свою подборку изображений в порядке.

Через пару минут вы увидите исполняемый файл *Findimagedupes* в поддиректории `bin` внутри корня Go. Чтобы приложение проанализировало файлы и показало дубликаты (при наличии таковых), просто передайте путь к своим изображениям как аргумент для *Findimagedupes*. Используйте опцию `-t <int>` для контроля сходства, где `<int>` должно быть в диапазоне 0–64, таким образом:

```
$ findimagedupes -t 22 ~/Images
```

Findimagedupes запоминает все хэши, и после того, как в первый раз пережует все ваши файлы, работает очень быстро.

Менеджер файлов

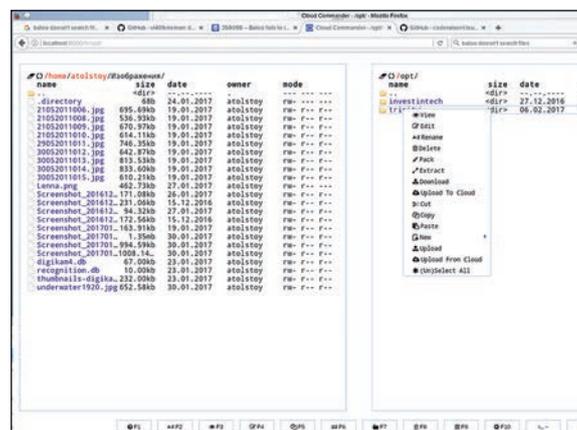
Cloudcmd

Версия: GIT Сайт: <http://bit.ly/2m6dLwP>

Не так давно мы восхищались амбициозным менеджером файлов *Sprut.io* (LXF208), который выглядит приложением из будущего — по крайней мере, если вы верите, что все интерфейсы в конечном итоге прилепятся к web-приложениям. *Sprut.io* был очень хорош для просмотра локальных и удаленных файловых систем, но следует признать: на сервере-то развертывание приложения из контейнера Docker было хорошей идеей, но на настольном ПК выглядело странно. На сей раз мы обнаружили похожее решение, с теми же преимуществами, но без подобных препятствий.

Cloudcmd — менеджер файлов, не без парочки бонусов. Во-первых, он может работать на любой системе, где есть графический web-браузер, и умеет обрабатывать современные пакеты Node.js через NPM. Во-вторых, он создан в проверенной временем двухпанельной парадигме, которой уже не один десяток лет.

Cloudcmd показывает вам ваши файлы почти так же, как приложения типа *Midnight Commander*. Вы можете переходить из одной панели в другую по клавише Tab и использовать те же привязки к клавишам, что и в других командерах (F5 для копирования, F8 для удаления, и т.д.), просматривать изображения, распаковывать архивы и выполнять множество других действий. Чтобы вывести контекстное меню с доступными функциями, щелкните правой кнопкой по любому месту панели. Помимо всего прочего, можно загружать локальные файлы или скачивать файлы из Сети, указав их URL. Нижняя панель с функциональными кнопками также предлагает отдельную кнопку для вывода встроенной консоли



➤ Наконец-то у нас есть web-приложение с замечательным откликом, способное конкурировать с *Midnight Commander*.

и удобную кнопку «чата», которая должна дать вам возможность связаться с командой разработки (если там кто-то есть).

Cloudcmd невероятно быстр и очень удобен в работе. Если вы создадите для него индивидуальный профиль web-браузера и скроете все относящиеся к браузеру кнопки управления и панели инструментов, будет трудно догадаться, что *Cloudcmd* является web-приложением. Установка *Cloudcmd* очень проста: запустите сперва `sudo npm i cloudcmd -g`, а затем серверную часть — по `cloudcmd`. И вы получите доступ к менеджеру файлов, перейдя на <http://localhost:8000> в любимом браузере.

Cloudcmd невероятно быстр и очень удобен в работе.

Док приложений

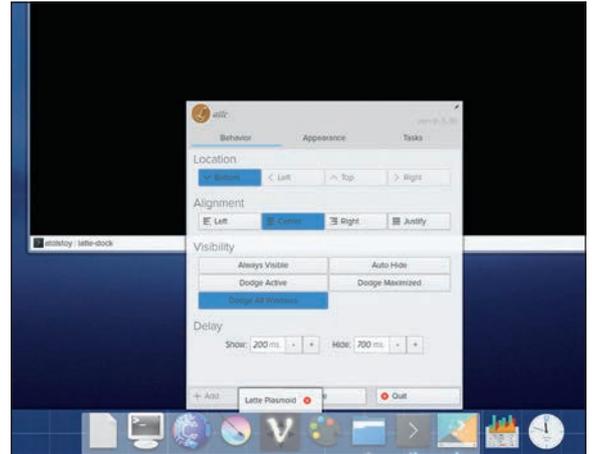
Latte Dock

Версия: 0.5.91 Сайт: <http://bit.ly/2mF7YBG>

Одна из неприятных проблем рабочего стола Plasma 5 — нехватка высококачественных виджетов рабочего стола (они же — плазмоиды). Последний термин кажется более правильным, поскольку стандартные компоненты рабочего стола, такие, как панель или меню приложений тоже являются плазмоидами, т.е. небольшими приложениями, написанными на языке быстрой разметки Qt (QML). В store.kde.org плазмоидов немало, но по-прежнему нет достойного дока для запуска приложений и догляда за работающими задачами. На данный момент есть две попытки заполнить этот пробел: *Now Dock* и *Latte Dock*, из которых второй, по нашему мнению, является более жизнеспособной и достойной опцией.

Идея дока приложений существует с тех пор, как Apple в начале 2000-х представил OS X. Панель с крупными значками занимает нижнюю часть экрана и может автоматически скрываться, что весьма разумно и экономит пространство экрана

по вертикали. Кроме того, панель предлагает отличную анимацию масштабирования, которая заставляет вас снова и снова наводить мышью на док. *Latte Dock* — классический плазмоид с набором настроек, благодаря которым он работает практически так же, как док Apple. Док отображает значки запущенных приложений, но чтобы значки оставались после остановки приложения, надо щелкнуть по каждому значку правой кнопкой и выбрать опцию Show Launcher When Not Running [Показывать программу запуска в выключенном состоянии]. *Latte Dock* также содержит закрепленные аналоговые часы, которые при щелчке по ним выводят стандартный виджет календаря Plasma. Отличное свойство *Latte Dock* — изобилие его настроек (щелкните



► Украшения только для KDE с эффектом плавного масштабирования и хорошей интеграцией с рабочим столом.

по плазмоиду правой кнопкой, чтобы получить к ним доступ), которые включают масштабирование, анимацию, отсрочку и режим работы. Он хорошо интегрируется с остальным рабочим столом, поддерживает Plasma 5 Activities и может показывать поверх значков информацию о состоянии (например, копирование файлов). *Latte Dock* — рекомендуемый и полнофункциональный аналог таких приложений, как *Docky* для рабочего стола Plasma 5. Вы можете найти его через стандартную функцию 'add new stuff [добавить новые программы]' в Plasma 5, которая переносит новые плазмоиды прямо на ваш рабочий стол.

Отличное свойство Latte Dock — изобилие его настроек.

Расширение Gnome Shell

Argos

Версия: GIT Сайт: <https://github.com/p-e-w/argos>

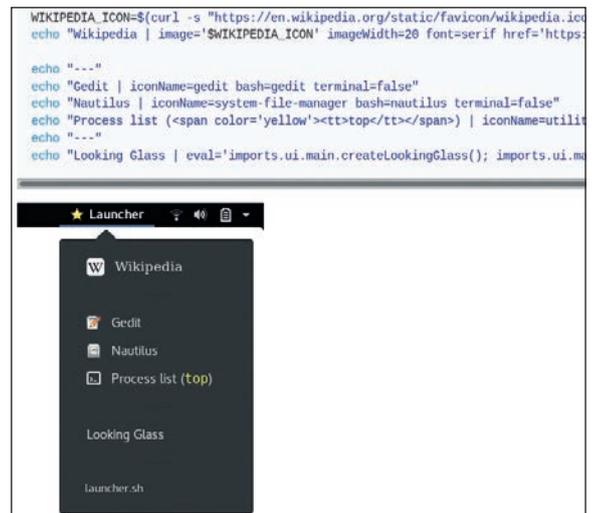
Те, кто использует Gnome Shell в качестве рабочего стола по умолчанию, оценят элегантность сайта <http://extensions.gnome.org>, который поставяет онлайн-контент из браузера прямо на ваш рабочий стол. А как насчет создания своего индивидуального расширения Shell? Для этого незачем быть разработчиком программ: поможет превосходная утилита *Argos*. Она является расширением Shell (добавляет выпадающую кнопку на верхнюю панель в Gnome Shell), но предназначена для превращения скрипта *Bash* хотя бы с одной командой Linux в расширение рабочего стола. Так, можно применить *Argos* для отображения сетевого трафика, свободного места на диске, погоды... всего, к чему обычно получают доступ через терминал, в качестве графического индикатора рабочего стола. Вам надо только подготовить скрипт *Bash*, производящий читаемый результат, и перекинуть его в `~/config/argos`. Для каждого скрипта *Argos* создаст свою кнопку

на верхней панели. В зависимости от вашего скрипта, *Argos* отобразит результаты прямо на панели или покажет какую-то полезную информацию во всплывающем окне при нажатии на индикатор. Вы даже можете создавать списки опций внутри индикатора, с помощью команды `echo`. Например, следующая простая команда приведет к запуску отдельного текстового редактора внутри вашего индикатора:

```
echo "Gedit | iconName=gedit bash=gedit terminal=false"
```

Официальная страница *Argos* на GitHub приводит ряд простых примеров, которые здорово помогают понять, как разместить свои функции на верхней панели своего Gnome Shell. А еще *Argos* поддерживает

Разместить свои функции на верхней панели Gnome Shell.



► Отличный инструмент компенсации скромных возможностей настройки Gnome.

плагины из *BitBar*, настройщика панели Mac OS X. Множество таких плагинов имеется на <https://github.com/matryer/bitbar-plugins> — для настройки времени, сети, музыки, новостей и кучи других источников информации. Многие плагины отлично ладят с *Argos* (если их код не заточен чисто под OS X) и дают вам еще большую свободу.

Для запуска *Argos* надо распаковать архив с расширением в `~/local/gnome-shell/extension` и включить соответствующий оператор в *Gnome Tweak Tool*.

HotGames Развлекательные приложения

Оборона башен

Machines vs machines

Версия: 1.3.2 Сайт: <http://bit.ly/2mKDLRY>

Всё, что вам нужно, чтобы заставить это приложение Ubuntu для телефона на QML запускаться на ПК — это среда Ubuntu SDK и немного силы убеждения. В него вполне можно играть с помощью мыши.

Вы должны защищать оконечность тропы (отмеченную сердечком) от постоянно продвигающейся вперед вереницы ботов. Они намерены достичь вашего сердца, а вы должны уничтожить их, разместив укрепленные башни вдоль тропы. За каждый разрушенный бот вы зарабатываете доллары и можете обновлять свои башни или потратиться на разблокировку более продвинутого оружия. В игре много видов боевых башен: некоторые замедляют боты, некоторые уничтожают их выстрелами или смертоносными волнами.

Разнообразие объектов превращает игру в сложный и динамичный квест, где

надо быстро принимать решение, например, об усилении имеющихся боевых башен или о продаже нескольких из них, чтобы разместить в других местах более мощные. Ключ к победе — оптимальное и сбалансированное планирование обороны. На каждом уровне вам дается десять жизней, и если вы допустили к своему сердцу не более девяти ботов, то можете получить следующие десять. Но самая захватывающая задача состоит в том, чтобы идеально защитить свою тропу и не позволить прорваться никому.

По завершении каждого уровня вы набираете звезд. Чем больше у вас звезд,



➤ Одна из лучших игр для телефона Ubuntu — на настольном ПК.

Защитить свою тропу и не позволить прорваться никому.

тем более мощное оружие вы можете купить во время следующего матча. Пакет по умолчанию предлагает 25 уровней, этого более чем достаточно, чтобы провести неделю или две, разрабатывая идеальные оборонительные решения против тупых ботов.

Доставка товаров

404 Elm street

Версия: GIT Сайт: <http://bit.ly/2niT9lq>

Это приятный способ провести свободный час и отличный выбор для любого пользователя Linux, который не хочет застрять в серьезной и требующей времени игре Steam. Изначально она была разработана в качестве забавного заполнителя места для страницы «Страница не найдена» сайта Zalando. Zalando — немецкая марка обуви и одежды, и в игре вы пытаетесь доставить их клиентам Zalando, чтобы доказать, что вы — старательный и расторопный заландист.

Экран показывает два модных магазина и четырех клиентов — полуголых людей, которые выражают свои мечты в облачках с мыслями. Вы должны нагрузить велосипед желаемыми товарами и доставить их каждому покупателю. Игра заключается в стратегии и планировании, потому что надо всё заранее продумать и решить, в каком порядке

посетить клиентов. Вы можете погрузить на свой велосипед максимум четыре товара, а чтобы куда-то добраться, нужно время, причем пока вы мчитесь к клиенту, тот может передумать и изменить свой заказ. Если вам не удастся обслужить клиента вовремя, тот начинает протестовать и орать (выглядит это довольно гадко, прямо как в жизни), а потом через некоторое время исчезает, и его заменяет другой человек — тогда вы теряете одну жизнь из первоначально полагающихся трех.

404 Elm Street — приложение СайтGL для работы в web-браузере, написанное на языке Elm, который предназначен для компиляции в высокопроизводительный

Если не обслужить клиента вовремя, тот начинает орать.



➤ Женщины возвращают купленную одежду и требуют возврата денег чаще, чем мужчины. И в игре тоже.

код JavaScript. Как мы уже говорили, в игру можно играть онлайн в магазине Zalando или на <http://zalando.github.io/elm-street-404>.

Вы можете запустить локальную копию игры, если на вашем компьютере есть Elm. Скачайте Elm с <http://elm-lang.org/install>, затем запустите команду `$ elm reactor` в директории игры и, наконец, перейдите на <http://localhost:8000/src/Main.elm> в своем браузере, чтобы увидеть игру.

Инструмент резервного копирования

GRSync

Версия: 1.2.6 Сайт: <http://bit.ly/2mndHKO>

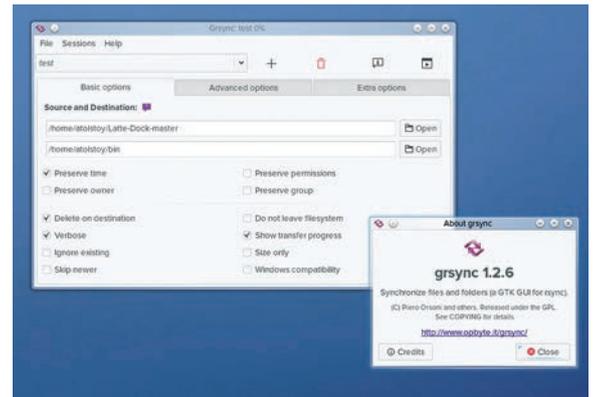
Инструменты резервного копирования делятся на несколько категорий — для дома и предприятия, полнофункциональные и минималистичные, графические и для командной строки. Плюс к этому разнообразию люди норовят подразумевать под термином «резервное копирование» разные вещи, так что вряд ли на свете существует решение резервного копирования, которое устроило бы всех.

Обратившись к Сравнениям инструментов резервного копирования в LXF207 и LXF138, мы там обнаружили, что *Arcata Backup* и *Deja Dup* победили всех своих конкурентов, но всё же далеки от совершенства. Например, когда мы применили *Deja Dup* для создания копии нашей двухтерабайтной директории `~/home` на внешнем жестком диске USB 3, это было невероятно медленно. Мы терпеливо ждали пару дней, отмечая, как приложение терзает жесткий диск, сжимая каждый файл и делая записи для будущих инкрементных изменений, и в конце концов плюнули и сдались.

В этом случае нам определенно нужно более примитивное решение, которое бы синхронизировало исходную и целевую директории без всяких издержек.

GRsync оказался для этого идеальным инструментом. Как и намекает его название, *GRsync* — это интерфейс к *Rsync* на *GTK*, в основном идентичный *LuckyBackup* — GUI для *Rsync* на *Qt*. Использовать *GRsync* очень просто: создайте сессию (разновидность профиля), укажите исходную и целевую директории, отметьте нужные окошки во вкладках Базовых и Расширенных опций, и всё.

GRsync позволяет посмотреть, какая команда *Rsync* будет выполнена (нажмите `Alt+R`), и, возможно, исправить ее до запуска. В приложении есть окна-флажки



Аккуратный и, возможно, самый здоровый интерфейс на *GTK* для *Rsync*.

для почти всех значимых опций командной строки *Rsync*. Например, можно сохранять файлы целевой директории, если она не пуста (отметьте `Make backups` [Создать резервную копию]) или переносить идентичную копию источника (отметьте `Delete on destination` [Удалить в точке назначения]), и т.д.

GRsync не имеет специальных функций интеграции с рабочим столом и, разумеется, не смог бы превзойти более профессиональное решение для резервного копирования, но это определенно хорошая опция для нишевого использования, если вам просто нужно наскоро сохранить несколько файлов на внешнее устройство или удаленную файловую систему.

Есть окна-флажки для почти всех значимых опций команд *Rsync*.

Менеджер паролей

Buttercup

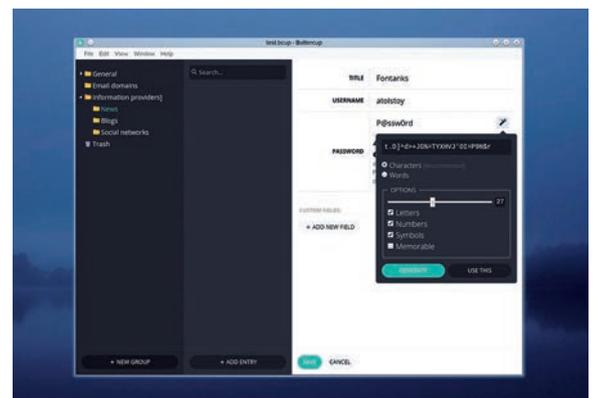
Версия: 0.9 Сайт: <https://github.com/buttercup/buttercup>

Не секрет, что наличие одинакового пароля для всех учетных записей онлайн — на редкость плохая идея. Однако большинство людей не могут запомнить десятки сильных паролей для своих учетных записей, включая электронную почту, форумы, социальные сети, обмен мгновенными сообщениями и прочие сервисы. *Buttercup* — это менеджер паролей, который пытается поддержать очень высокий уровень безопасности, не усложняя жизнь пользователя.

Это симпатичное приложение на *JavaScript*; оно предлагает удобный интерфейс для организации ваших паролей в своего рода хранилище, защищенное мастер-паролем. Каждое хранилище называется в *Buttercup* «архивом», и у вас может быть сколько угодно архивов. Внутри каждого архива можно создавать древовидную структуру категорий и заполнять их. *Buttercup* позволяет сохранять вместе с именами и паролями напоминания и подсказки —

в каждую запись можно добавить любое количество индивидуальных полей. Найдя слабый пароль, приложение показывает соответствующее уведомление. Значок с волшебной палочкой справа от поля ввода пароля размещает всплывающий генератор паролей с расширенными опциями прямо у вас под рукой.

Buttercup сопровождается расширением браузера с тем же названием для *Chrome* и *Chromium*, а значит, вам не надо вручную копировать и вставлять свою аутентификационную информацию из настольного приложения в браузер. Технически *Chrome* и *Chromium* (и большинство других браузеров) предоставляют те же функции по умолчанию, но вы, возможно,



В *Buttercup* нам особенно понравился его продуманный генератор паролей.

захотите воспользоваться сторонним менеджером паролей, если доверяете ему больше. *Buttercup* кажется весьма надежной заменой встроенным менеджерам паролей благодаря его расширенным функциям, таким, как поиск, корзина, генератор паролей и т.д.

Всё решение очень надежное: архив после сохранения шифруется по алгоритму `AES 256bit CBC` с помощью `SHA256 HMAC`. Шифрование выполняется, как только пароль будет подготовлен и «подсолен» `PBKDF2` за 1000 итераций. Это безопасно. LXF

Buttercup позволяет сохранять вместе с паролями подсказки.

На диске

Дистрибутивы, приложения, игры, книги и всякое-разное...

Лучшее из Интернета, упакованное в 8 ГБ качественного DVD.



Дистрибутивы

Недавно у меня (по моей же вине) произошла мелкая катастрофа. Не буду утомлять вас деталями, но я сдуру запустил эквивалент `rm -fr /` невзирая на то, что текущие версии `rm` защищают от подобного идиотизма, и стер часть своей ОС, прежде чем смог это тормознуть! К счастью, я старательно делаю резервные копии, поэтому восстановил свою файловую систему `root` из вчерашней резервной копии и решил, что всё будет хорошо. Ан нет: у меня возникли всяческие проблемы, с виду непричастные. Установка ряда программ завершилась несуразным сообщением об ошибке, и я не смог сделать распечатку и кучу других вещей, которые мой мозг просто отторгнул, дабы сохранить мой рассудок.

Я попытался проработать каждую проблему, выясняя причины — нашлись не все, и я, наконец, решил воззвать к помощи: составил письмо в список рассылки, вбил в него всю сопутствующую информацию... и тут меня осенило. Немного порывшись, я нашел решение! Иногда надо перестать искать срочное решение и просто сосредоточиться на проблеме — и решение появится. В чем же была проблема? В восстановленной копии не хватало разрешения группы `x` в некоторых, с виду произвольных директориях.

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 Linux для творческих личностей

Ubuntu Studio 16.04.2

Дистрибутивы Linux многогранны — вы можете начать с той же базы, но в итоге получить совершенно иные конечные продукты, что ярко демонстрируют два предложения этого месяца. Имя Ubuntu Studio — неплохая подсказка и насчет его происхождения, и насчет его цели. Это дистрибутив на базе Ubuntu, «заточенный» на применение ПК для творчества. В нем куча программ для аудио, графики, видео, фотографии и публикаций. Да, можно установить стандартную систему Ubuntu и затем добавить требуемые программы, но зачем вам эта морока и труды по притиранию их друг к другу, если кто-то уже всё сделал за вас? Так, пользователи аудио оценят тот факт, что Jack уже установлен и настроен.

Приложения для обработки видео требуют большой мощности процессора, а аудиоприложения — системы с хорошим откликом, особенно для работы в реальном времени. Мощность современных настольных ПК до смешного избыточна для обычных областей применения, но есть и исключения, и это именно оно. Далее надо суметь точно настроить свою систему, выжав из нее максимум, и львиная доля этой работы уже сделана за вас Ubuntu Studio.

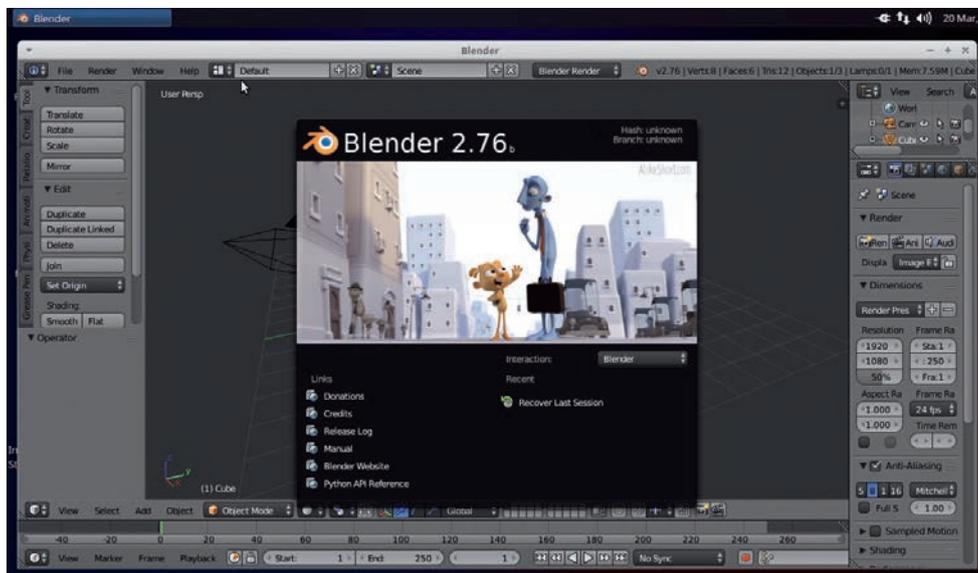
Аналогично стандартным релизам Ubuntu, есть релизы с долгосрочной поддержкой (LTS) и промежуточные. Если вы настраиваете рабочую станцию для создания творческого материала, то стабильная платформа обычно важнее наличия наивысшего номера версии программы, поэтому у нас тут стабильный LTS-



В Ubuntu Studio масса приложений для работы с аудио, и не меньший выбор другого ПО для творчества.

релиз, с поддержкой не менее четырех лет. Как и с другими дистрибутивами на этом DVD, вам не должна понадобиться аутентификационная информация ниже. Все дистрибутивы настроены на вход в среду рабочего стола как пользователь `live`. Если вы видите окно-приглашение, значит, в этом процессе произошел сбой; наиболее вероятная его причина — проблема с драйверами к вашей видеокарте. Загляните в FAQ на DVD, там есть более подробная информация. Другая возможная причина запроса пароля — настройка скрин-сейвера на блокировку экрана. Для дистрибутивов с пустым паролем просто нажмите Enter для разблокировки.

Аутентификационная информация: имя пользователя — `ubuntu-studio`, пароль пустой.





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.

Очень легковесный Linux

XenialDog 2017-01-22

Puppy Linux — давний, очень легковесный дистрибутив. A Dog Linux — пример того, что бывает, когда Puppy вырастает, матерееет, но остается поджарым [англ. Dog — собака, Puppy — щенок]. Есть вариант Dog Linux на базе Debian, но наш-то основан на Ubuntu, Xenial Xerus. XenialDog имеет опцию постоянного сохранения: если у вас есть раздел с меткой casper-gw или файл с именем casper-gw в корне раздела жесткого диска, то все изменения запомнятся там. Аутентификационная информация: имя пользователя и пароль — **puppy**.



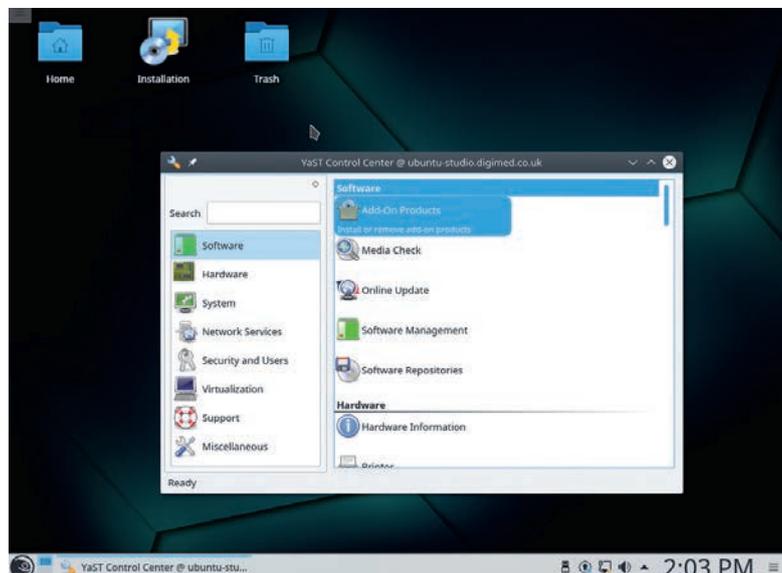
Постоянно обновляемая ОС

openSUSE Tumbleweed

Есть два дистрибутива openSUSE. Мы ранее включали ветвь Leap, которая следует традиционной схеме, основанной на выходе релизов, и постоянно требует переустановки или, как минимум, обновления. Вдобавок вы должны ждать следующего релиза, чтобы получить все свежие мелкие обновления. Tumbleweed, который вы найдете на DVD этого месяца, является дистрибутивом с возобновляемым релизом. Т.е. у него нет отдельных версий релизов — новые версии каждой программы появляются, как только они станут доступными.

Это может означать обновления немного чаще, зато вам никогда не понадобится полный пересмотр системы — версия дистрибутива всегда самая последняя. Тут есть плюсы и минусы. Хорошо то, что у вас всегда самые свежие программы; но это и плохо, ведь они еще не вполне протестированы. Если вам нужен прочный стабильный дистрибутив, поищите в другом месте; но чтобы всегда иметь свежечок, берите Tumbleweed.

Аутентификационная информация: имя пользователя — **linux**, пароль пустой.



» Будьте лучшим с Tumbleweed.

И еще!

Системные инструменты

Главное

Checkinstall Установка tar-архива с помощью менеджера пакетов.

GNU Core Utils Основные утилиты, которые обязаны присутствовать в каждой операционной системе.

Hardinfo Инструмент тестирования системы.

Kernel Исходный код последнего стабильного релиза ядра.

Mentest86+ Проверьте ОЗУ на предмет неисправностей и сбоев.

Plop Простой менеджер загрузки для запуска ОС с CD, DVD и USB.

RaWrite Создавайте загрузочные диски в MS-DOS в Windows.

SBM Независимый от ОС менеджер загрузки с несложным интерфейсом.

WvDial Соединяйтесь с Интернетом через телефонный модем.

Чтение

Книжная полка

Расширенное руководство по скриптам Bash Изучите написание скриптов еще лучше.

Руководство Bash для начинающих Овладейте написанием скриптов Bash.

Руководство по созданию скриптов Bourne Shell Начните осваивать скрипты оболочки.

The Cathedral and the Bazaar [Собор и Базар] Классический текст Эрика С. Реймонда [Eric S Raymond], объясняющий преимущества открытой разработки.

Справочник администратора Debian Базовое руководство для системных администраторов.

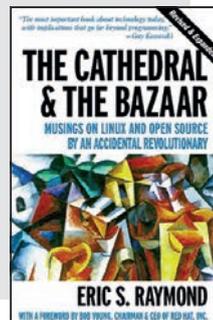
Введение в Linux Удобное руководство со множеством подсказок для начинающих пользователей Linux.

Словарь Linux Терминология Linux от А до Я.

Linux Kernel in a Nutshell [Ядро Linux в Двух Словах] Введение в ядро, написанное великим магистром ядра Греггом Кроа-Хартманом [Greg Kroah-Hartman].

Руководство системного администратора Linux Контролируйте свою систему.

Обзор инструментов Полный обзор инструментария GNU.



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!

LXF222 Апрель 2017



250 руб.

- » **Стать Linux-джедаем** Изучаем терминал
- » **В Интернет — невидимкой** Дистрибутивы конфиденциальности
- » **Серверы на все лады** Два учебника и статья
- » **Сайт с галереей** Адаптивный web-дизайн

LXFDVD: Devuan, Linux Mint, ROSA, Scientific Linux, Ubuntu Budgie, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_222/
PDF-версия: shop.linuxformat.ru/elxf_222/

LXF221 Март 2017



250 руб.

- » **Прощай, Google!** Мы найдем лучше
- » **Ваш добрый демон** Выбор ОС BSD
- » **Сервер на халяву** CentOS, клон Red Hat
- » **Linux + Windows** Сожительство возможно

LXFDVD: ExTiX, Porteus, Rebecca Black Linux, Siduction, Ubuntu, Zorin, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_221/
PDF-версия: shop.linuxformat.ru/elxf_221/

LXF219/220 Февраль 2017



250 руб.

- » **Linux навсегда** А Windows — на мыло
- » **Дистрибутивы в весе пера** Легковесный не значит убогий
- » **Графика будущего** Переходим на Wayland
- » **Рация-переноска** Ностальгируем с Raspberry Pi

LXFDVD: antiX, Bodhi, Fedora, Manjaro, openSUSE, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_219-220/
PDF-версия: shop.linuxformat.ru/elxf_219-220/

LXF218 Январь 2017



250 руб.

- » **Приручение Яка** Он же — Ubuntu 16.10
- » **Новые, хромовые** Выбираем хромбук
- » **Помирать нам рановато** Шанс 32-битных машин
- » **Телефон в терминале** Управляем с Linux-ПК

LXFDVD: BunsenLabs, Calculate Linux, GeckoLinux, PeiScanHPC, Ubuntu, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_218/
PDF-версия: shop.linuxformat.ru/elxf_218/

LXF217 Декабрь 2016



250 руб.

- » **Сплошная малина** Всё про Raspberry Pi
- » **Болтунам** Без находок для шпиона
- » **Благодеяния** Свежая WordPress
- » **Интернет вещей** Строим домашних, построим и дом

LXFDVD: Android-X86, Elementary OS, Korora, Mint KDE, Maui, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_217/
PDF-версия: shop.linuxformat.ru/elxf_217/

LXF216 Ноябрь 2016



250 руб.

- » **Береженого и Бог бережет** Взлому — не быть!
- » **Какая штука...** Чем печатать 3D-объекты
- » **Кофеварка на Raspberry Pi** Рулим температурой
- » **Блог на Phoenix** И пускай комментируют

LXFDVD: 4MLinux, Kubuntu, Lubuntu, Xubuntu, Mint 18 Xfce, OpenMandriva Lx 3.0, SELKS 3.0, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_216/
PDF-версия: shop.linuxformat.ru/elxf_216/

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

» Содержание

LINUX
FORMAT

Страница 1

ДИСТРИБУТИВЫ

Ubuntu Studio 16.04.2 (64-битный)
Дистрибутив на базе Ubuntu для творцов аудио, графики, видео, фотографий и публикации. Содержит уже настроенные приложения в соответствии с этой целью.

openSUSE Tumbleweed (64-битный)
Дистрибутив с возобновляемым релизом.

Xenialdog 2017-01-22 (32-битный)
Легковесный дистрибутив на базе Ubuntu (Kernal Xerus).

СРАВНЕНИЕ: ПРОГРАММЫ САД

FreeCAD
LibreCAD
OpenSCAD
Ocad
SolveSpace

НОТРИКС

404 Elm Street (Git) Игра: доставка товаров
<http://bit.ly/2z1l19d>
Adios (Git) Расширение GNOME Shell
<https://github.com/r-e-w/adios>

УЧЕБНИКИ

Android Studio
Python Find
Coovs2D
<https://github.com/Seelock/Droid>

Azrainger 2.0.2

Редактор изображений
<http://bit.ly/2z2w0RN>
Winetrip 0.9 Менеджер паролей
<https://github.com/Winetrip/Winetrip>
Cruciald (Git) Менеджер файлов
<http://bit.ly/2z2vdlw>

Findimagepress

(Git) Поиск изображений-дубликатов
<http://bit.ly/2z2w0RN>
gVSync 1.2.6 Инструмент резервного копирования
<http://bit.ly/2z2w0RN>
Knipr 1.3 Ножины
<http://bit.ly/2z2w0RN>

Late Dock 0.5.97

Док приложения
<http://bit.ly/2z2w0RN>
Machines vs machines 1.3.2 Игра: оборона башен
<http://bit.ly/2z2w0RN>
Ornav 0.4 Сканер QR-кода
<https://github.com/Seelock/Droid>

Описание на обороте »

Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials [Главное]» на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя_программы-1.0.1.i386.rpm** — вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя_программы-1.0.1.i386.deb** — такой же пакет, но уже для Debian;
- » **имя_программы-1.0.1.tar.gz** — обычно это исходный код;
- » **имя_программы-1.0.1.tgz** — тот же файл, что и выше этажом по списку: “tgz” — это сокращение от “tar.gz”;
- » **имя_программы-1.0.1.tar.bz2** — тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** — также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** — двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** — двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** — версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Внимательно прочтите это перед тем, как использовать LXF DVD!

LIVE-ДИСК: ЗАГРУЗИ > ОПРОБУЙ > УСТАНОВИ!

Ubuntu 64-БИТНЫЙ Studio 16

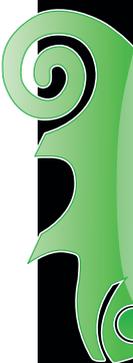
Идеальный открытый комплект для создания мультимедиа

- » Обработка фотографий в формате Raw
- » 3D-рендеринг, редактирование видео
- » Средства микширования звука для профи



openSUSE Tumbleweed

С поддержкой уровня предприятия



64-БИТНЫЙ

Содержание



ПОМОЩЬ

Руководство новичка
 Ответы
 Чью (FAQ)

ГЛАВНОЕ

Checkinstall
 Coreutils
 Hardinfo
 Kernel
 Memtest86+
 Plor
 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 [Greg Kroah-Hartman]
Rule 3 Знаменитый справочник
System Administrators Guide Руководство по базовому администрированию Linux
GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

Страница 2

ДИСТРИБУТИВЫ

ferret OS (64-битный)
 Настольный дистрибутив Linux на базе Mint. Настраивается с рабочим столом Cinnamon, включает уровень совместимости WINE для запуска приложений Windows, а также ПО для повышения производительности WFS, в основном совместимое с Microsoft Office и веб-браузером Vivaldi.
IPFire 2.19 — Core Update 110 (32- и 64-битный, без live-режима)
 Дистрибутив Linux, ориентированный на простоту настройки, хорошую обработку и высокий уровень безопасности и расширяемый различными дополнениями. Управляется через интуитивно понятный веб-интерфейс со множеством опций настройки для новичков и опытных сидадминов. Поставляется с менеджером пакетов Pakfire.

RescueOS 1.0.0 (64-битный)
 Крошечный дистрибутив Linux, который управляет всей ОС как контейнером Docker, включая системные службы, такие как `udev` и `rsyslog`. Содержит минимум ПО, необходимого для запуска Docker, благодаря чему объем его загрузки очень мал. Все остальное можно добавить динамически через Docker.
Rescatus 0.41 Beta 1 (32- и 64-битный)
 Дистрибутив на базе Debian, с графическим мастером для спасения установок GNU/Linux после сбоя. Варианты восстановления включают исправление загрузчика Grub после установки Windows, сброс пароля Linux и Windows и проверку файловой системы Linux.

Все дистрибутивы представлены ISO-образом, который можно записать на отдельный носитель, и загрузить в live-режиме прямо с LXF DVD. У всех присутствует возможность установки на жесткий диск.

**Пожалуйста, перестаньте использовать
 имя Адного Диска озанковать
 с инструкцией, опубликованной
 в журнале на стр. 109!**

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕФЕКТИВНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на вашем диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех этапах производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аналитический сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, представленных на нем программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, посетите www.linuxformat.ru для получения информации о всех наших партнерах и программах.

Тираж издательства ООО «Марком», 188652, Россия, Ленинградская область, Всеволожский р-н, дер. Юрки, Школьная ул., 7-а. Лицензия ИМПР. ВАР N 77-03.

Создание установочных дисков при помощи cdcrecord

Самый быстрый способ записать ISO-образ на чистую матрицу — это обратиться к программе `cdcrecord`. Для всех перечисленных ниже действий потребуются права суперпользователя-`root`. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdcrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке — например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdcrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование `cdcrecord`, сохраните некоторые настройки в файле `/etc/default/cdcrecord`. Добавьте по одной строке для каждого устройства записи (скорее всего, в вашей системе присутствует только одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке — метка; затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Для записи ISO-образа вам осталось набрать команду

```
cdcrecord -v /path/to/image.iso
```

Если вы не принадлежите к любителям командной строки, в таком случае вам придет на помощь утилита `gcombust`. Запустите ее от имени `root` и выберите вкладку `Burn` и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажмите на `Combust!` Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Использовать Linux для записи компакт-диска не обязательно. Программы вроде `cdcrecord` просто переносят двоичные данные на чистую матрицу. Все необходимые файлы уже включены в ISO-образ, который распознается любой операционной системой, будь то Linux, Windows, Mac OS X или AmigaOS.

Нет устройства для записи дисков?

Если у вас нет устройства, с помощью которого можно было бы записать образ на диск, можно найти какого-нибудь друга или организацию, у кого есть компьютер с дисководом, и прожечь диск у них. Опять-таки, вам подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт разработчика дистрибутива.

LINUX FORMAT

Подписывайтесь и читайте Linux Format на iPad или iPhone!

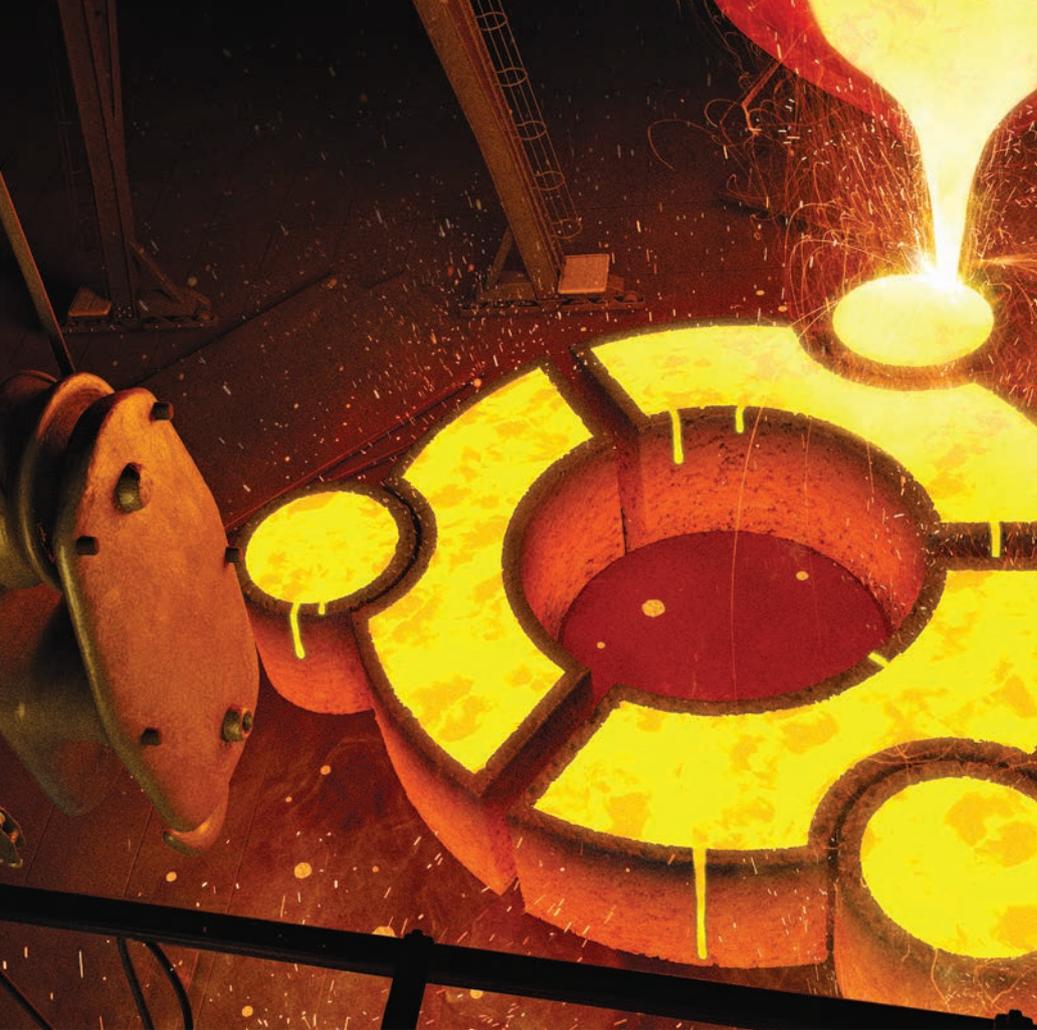
Доступно
в AppStore



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

 zinio™
Доступно в Google Play





LINUX FORMAT

Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № ФС77-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж печатной версии 1200 экз., распространение электронной версии 30 000 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 15914

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор [Neil Mohr] neil.mohr@futurenet.com

Научный редактор Джонни Бидвелл [Jonni Bidwell]

jonni.bidwell@futurenet.com

Выпускающий редактор Крис Торнетт [Chris Thornett]

chris.thornett@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоза

[Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Матс-Таге Аксельссон [Mats Tage Axelsson], Джонни Бидвелл [Jonni Bidwell], Нейл Ботвик [Neil Botwick], Стюарт Бернс [Stuart Burns], Нейт Дрейк [Nate Drake], Кент Ельчук [Kent Elchuk], Джон Найт [John Knight], Джон Лэйн [John Lane], Лес Паундер [Les Pounder], Фил Савидж [Phil Savage], Зак Стори [Zak Storey], Крис Терстен [Chris Thursten], Александр Толстой [Alexander Tolstoy], Михалис Цукалос [Mihalis Tsoukalos], Евгений Балдин, Михаил Вознесенский, Андрей Гондаренков, Алексей Федорчук, Максим Черепанов, Лада Шершова

Иллюстрации Шейн Коллиндж [Shane Collinge]

Иллюстрация с обложки www.magicstorch.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 в виду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на диски — CD или DVD, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет 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.futurepic.com.

В июньском номере:

Угадали?

Ubuntu 17.04!

Чокнемся стаканами с лимонадом в ознаменование нового релиза Linux.

Тесты в Linux

Тестирование открытой системы, стресс-тесты и калибровка производительности — расскажем про всё.

Ярмарка Умельцев 2017

Живой репортаж прямо оттуда, со сплетнями про лучшие проекты и всю прочую потеху.

Соберем роутер

Как почему? Потому что можем! Скомбинируйте Ubuntu Server и старое оборудование для домашней маршрутизации.

Содержание будущих выпусков может меняться — вдруг мы заработаемся, составляя на них планы...



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

16+

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.