

DVD ДАРОМ!



СВЕЖИЙ UBUNTU 17.04

LINUX FORMAT

Главное в мире Linux

Июнь 2017 № 6 (224)

64 страницы
учебников
и статей!

- » Собираем роутер на Ubuntu
- » Ставим брандмауэр как профи
- » Продвинутом: профили терминала



Академия кодига:
Web-разработка с Django

UBUNTU 17.04 изнутри!

Вскрываем механизмы
судьбоносного релиза Zesty

- » Будущность Unity 8
- » AMD внутри — ядро 4.10
- » Преображенный Gnome 3.24

ЕСТЬ
**ЭЛЕКТРОННАЯ
ВЕРСИЯ**
ДЛЯ ВАШЕГО
ПЛАНШЕТА!



Ярмарка умельцев 2017

« Сообщество играет огромную
роль в этом движении,
все помогают друг другу »

Робин Хартли — о создании скоростной чудо-клавиатуры

Задать жару Linux

Сравнительные и стресс-
тесты, и не только!

Web-разработка

Готовим инструментарий

» Все необходимое, чтобы стать раз-
работчиком сайтов и web-приложений



Сравнение

Дистрибутивы KDE

» Подобный опыт настольного
Linux не встретишь больше нигде



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

Linux center
www.linuxcenter.ru





iTeleRadio

ИНТЕРАКТИВНОЕ РАДИО&ТЕЛЕВИДЕНИЕ

iteleradio.ru



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: что вы думаете о возврате Ubuntu на Gnome? Вас это радует, огорчает, или у вас Slackware и вам все равно?



Джонни Бидвелл

Мне немного грустно, хотя я точно не знаю, почему. Я пользуюсь Gnome каждый день и всегда находил Unity чуть более неуклюжим/громоздким. Видимо, я считал, что он дружелюбнее к новичкам, а теперь много людей, которые никогда не сталкивались с Gnome 3, переходят на 18.04 и вдруг оказываются в Terra incognita.



Нейл Ботвик

Я думаю, многие пользователи будут огорчены. Unity уже использует технологию Gnome 3, и по-моему, «новый» рабочий стол будет намного больше похож на Unity, чем старый рабочий стол Ubuntu. Если бы они выкатили Cinnamon, все было бы иначе. Но я упертый фанат KDE, и что я могу знать?



Лес Паундер

Когда Unity вышел впервые, я разлюбил Ubuntu. Долгие годы мне нравился рабочий стол Gnome, а до этого я был пользователем KDE. Но Unity для меня просто не работало. Прокрутим 2016–2017 годы — и теперь я использую Unity, но тайне рад возвращению Gnome.



Маянк Шарма

Вся эта история — вразрез с этикой открытого кода. Вот как насчет беспрестанного пения о достоинствах Mir и Unity 8 для настольных ПК? Просто потому, что Шаттлворт не сумел продать достаточное количество телефонов, он вдруг осознал, что фрагментация — это не здорово, и «прагматично» решил прикрыть проекты? Фи!



Шашанк Шарма

Когда я пишу обзоры и встречаюсь с неважным программным обеспечением, мне становится больно, потому что критиковать и отклонять чьи-то любовные труды непросто. Не будучи пользователем Ubuntu, я тоже испытываю боль из-за этого развития событий, тем более, что Unity отшвырывается столь поспешным образом.



Валентин Синицын

Я никогда не был сторонником Gnome, но история, как известно, развивается по спирали. Был Gnome, потом Unity, теперь снова Gnome — кто знает, что будет дальше? Я скучаю по временам, когда дистрибутивы предлагали KDE и Gnome, на выбор. Может, пора сделать Kubuntu магистральной версией? И не надо переписывать ничего на Qt.



Впервые и не скрываясь

» Месяц назад в Москве состоялась конференция разработчиков отечественных операционных систем — OS DAY 2017. Событие получилось, прямо скажем, неординарное. Наряду с широко известными проектами были представлены разработки, ранее известные только весьма узкому кругу специалистов, и совсем новые разработки. Перечислить все интересное в объеме этой колонки невозможно. Презентации см. на сайте конференции osday.ru по ссылкам под портретами докладчиков.

Мое личное впечатление от конференции двояко. Оригинальные разработки в стране есть, и они развиваются. Более того, далеко не все ОС базируются на готовых свободных проектах — есть свои, и весьма интересные решения. Большинство систем уже поддерживает аппаратные архитектуры x86, ARM и MIPS, а в планах — «Эльбрус», благо серийное производство компьютеров на его базе, по сообщениям прессы, началось. Это хорошая новость. Вот только системы эти в основном проприетарные. Оно и понятно: авторы даже не рассматривают возможность широкого, «гражданского» использования своих творений. А дальше — замкнутый круг. Нет сообщества разработчиков — нет приложений, нет приложений — нет внедрений, нет внедрений — нет смысла портировать софт на такую платформу... Все так и останется в узком секторе спецсистем и госзаказа. И это плохая новость...

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

«Ради Бога, остановитесь хоть на миг, перестаньте работать, оглянитесь вокруг.» Лев Толстой

Обзоры

Parrot Security OS 3.5 14

Мы протестировали этот дистрибутив для тестирования на вторжения и вообразили себя невидимками в Сети. Кхе...

4MLinux 21.0 15

Амбициозный минималистский дистрибутив — соблазнительная штука. Но будешь ли съят одним соблазном? Вот и попробуем разобраться.

FreeNAS Corral 16

Популярное решение для сетевого хранилища обретает новую жизнь: прежняя версия переработана снизу доверху.

Google WiFi 17

Всего один роутер?! Бесполезняк. Чтобы достучаться до всех закоулков Башен LXF, нужно по крайней мере три. Стены из батского песчаника уж очень толстые...



» А это случайно не баночка для крема?

Civilization VI 18

Собирайтесь, детки, в круг — мы вам поведаем о незапамятных временах, когда на месте космодрома колосились поля, много поколений назад.



» Посмотрим, как вы будете править своим крошечным королевством.

Внутри Ubuntu 17.04

Мы наложили лапы на свежий релиз Ubuntu 17.04: спихнули с него крышу и сунули нос вовнутрь, чтобы разобраться, почему он тикает. См. с. 28

Сравнение: Дистрибутивы KDE с. 22



Ярмарка умельцев 2017



« Это больше касается отношения к делу, готовности учиться и чувства общности »

Робин Хартли на главном шоу мастеров Британии с. 36

На вашем бесплатном DVD

LIVE-ДИСК: ЗАГРУЗИ > ОПРОБУЙ > УСТАНОВИ!

Ubuntu 17.04

Новинка-2017!

- » Ядро 4.10
- » Рабочий стол Unity 8
- » Тысячи приложений!

32- и 64-битный



Ubuntu 17.04 32-битный
Ubuntu 17.04 64-битный
Linux Lite 3.4 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

Пользователям Raspberry Pi



Новости Pi 90

Продажи Pi обогнали Commodore 64 (а скоро обгонят и всю его линейку), появились Mac Pi, и внедрен язык программирования Julia.

Asus Tinker Board 91

Вышел новый одноплатный компьютер от Asus — и размышлялся о победе над Pi.

Цифровой календарь 92

Соорудим себе настенный календарь на Raspberry Pi, чтобы стать самыми организованными в мире людьми.

Minetest 96

Minetest от Microsoft пора сливать. Выбираем Minetest с открытым кодом, для Pi и ПК.



Ищите в номере

Тестируем Linux 40

Тесты — штука хитроумная, но с нашим руководством все будет просто (или хотя бы проще).

Учебники

Основы терминала

Bash на Bash 46

Сравниваем файлы и углубляемся в историю Bash, чтоб сэкономить на вводе команд.

Основы терминала

Разные профили 48

С помощью профилей, на лету переключаем терминал в разные виды! Каждой задаче — свою цветовую схему.

Брандмауэры

pfSense 50

Оказывается, построить роутер с брандмауэром — штука вовсе не хитрая.

Построение графов

Graphviz 52

Прямо и не знаем, насколько спянные нами графы обворожительны... но мы любим наши графы...

Серверы

HAProxy 56

Для разнообразия позанимаемся настройкой обратного прокси — сбалансируем нагрузку на серверах.

Шифрование

GnuPG 60

Осознаем, кому доверять свои ключи, а кому нет, и что об этом думает GnuPG.

Сети

Соберем роутер 64

Роутеров на белом свете полно, но наш-то, несомненно, превосходит всех прочих по управляемости.

Шифрование

TrueCrypt 68

Водрузим поверх VeraCrypt удобную утилиту Java, которая зашифрует ваши файлы в три прыжка (шага).

Интернет вещей

Аякс-скриптоносец 70

Принимаемся управлять Послушным Домом через JavaScript и AJAX в браузере.

Встроенное

Роботы на micro:bit 74

Соберем двух радиоуправляемых роботов на BBC micro:bit, запрограммируем их на MicroPython и натравим друг на друга!

Академия кодига

Django на свободе 78

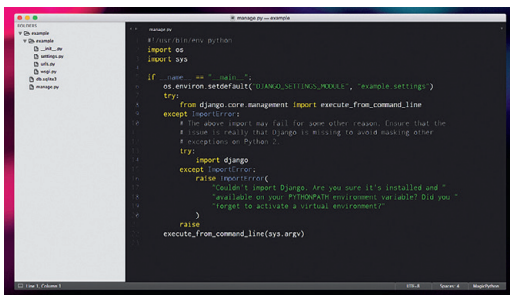
Разберемся с основами фреймворка Django и напишем в нем свои первые строки кода.

Web-разработка 82

Объясняем, как создать готовую к употреблению машину для web-разработки.

Шахматы на Python 86

Гроссмейстеров из нас не вышло; перекалифицируемся в программеры и изучим шахматный движок.



Постоянные рубрики

Новости 6

Google помогает ЖКХ своим IoT, процессоры AMD поднимают планку, российская САПР погружается в виртуальную реальность, WD расширяет память, IBM готовит квантовые процессоры, появились ПК на базе «Эльбрус», ноутбук Lenovo огорчает ценой, а Асег изогнула монитор.

Вести мобильных ОС 20

Chrome OS идет в ночное, ARM озабочилась чипами для искусственного интеллекта, планшет от HP обошел Microsoft, а Android OS угнездилась на автомобилях Volvo.

Сравнение 22

Новый Ubuntu — дело хорошее, но дистрибутивы с KDE ничуть не хуже: Chakra, KaOS, Manjaro, Maui, Netrunner.

Интервью LXF 36

Робин Хартли, Эдриан Мак-Юэн и Лоррейн Андервуд прибыли на Ярмарку умельцев и людей посмотреть, и себя показать.

HotPicks 100

Отведаете горяченького! Лучшие в мире новинки свободного ПО: Classifier, GNU Nano, Guetzli, LanguageTool, Man vs Olives, Mate,

Meteo-Qt, MTPaint, NTFS-3G, Tank Island, Webenginepart.

Диск Linux Format 106

Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108

А зря. Там было столько всего хорошего про новенький Raspberry Pi Zero W...

Через месяц 112

Переодеемся «белыми» хаке-рами и встаем на защиту ваших сетей от атак! Курс молодого бойца по тестированию на вторжения.



В ЭТОМ НОМЕРЕ: Облака для ЖКХ » AMD угодила всем » САПР с ВР » Новый SSD » IBM и кванты » Эльбрус взят » Имя ему Legion » Монитор на 35”

ИНТЕРНЕТ ВЕЩЕЙ

Коммунальные облака

Новый сервис Google предназначен для управления устройствами IoT и обработки генерируемых ими данных.



» Рубрику готовил
**АНДРЕЙ
ГОНДАРЕНКОВ**

Облачный сервис Google Cloud IoT Core обеспечивает предприятиям, в т.ч. службам ЖКХ и транспортным агентствам, безопасное подключение к облачной платформе глобально распределенных устройств Интернета вещей; данные централизованно управляются и интегрируются с аналитическими сервисами Google. 2 месяца тестировала новый сервис Energyworx, компания из 40 сотрудников, предоставляющая предприятиям ЖКХ услуги по аналитике данных и пользующаяся облачными сервисами Google с 2014 г.

Cloud IoT Core — это, по сути, коммуникационный мост между Google analytics

и устройствами IoT; многие из них применяют протоколы, преобразуемые сервисами Google без ручного вмешательства; с помощью сервиса коммунальщики могут отправлять на «умные» счетчики команды управления. Energyworx платит Google только за устройства IoT, используемые для получения данных, что снижает затраты: в целом, плата за облачные сервисы Google обходится компании в десятую часть от суммы, выплачиваемой за сервисы платформы Amazon Web Services, которыми Energyworx пользовалась до 2014 г. «Мощь облаков быстро растет, и этот подход [Google IoT] станет масштабируемым, —

говорит основатель и визионер Energyworx Эдвин Пут [Edwin Poot]. — Мы ничего не устанавливаем и не сопровождаем, мы просто программируем сервис и используем его, не беспокоясь ни о чем другом».

» Сервис Google Cloud IoT Core поможет бизнесу решать ресурсоемкую задачу сбора и анализа информации от многочисленных и разнообразных устройств IoT.



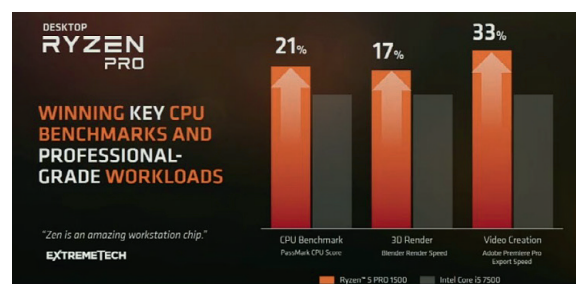
ПРОЦЕССОРЫ AMD

И для геймеров, и для бизнеса

Дополняя линейку процессоров Ryzen версиями Mobile и Pro, AMD пытается избавиться от репутации «очередного бюджетного чипмейкера».

Вслед за процессорами для настольных ПК Ryzen 3, 5 и 7, встретившими, по словам Джима Андерсона [Jim Anderson], старшего вице-президента и генерального менеджера AMD Client and Computing Group, «теплый прием у наших партнеров — изготовителей аппаратуры», во втором полугодии AMD выпустит ориентированный на модели 2-в-1 и традиционные ноутбуки процессор Ryzen Mobile, о чем было объявлено на 2017 Financial Analyst Day. Мобильный APU Ryzen обеспечит «фантастическое время работы от аккумулятора, прекрасную производительность в ресурсоемких приложениях и феноменальное для ноутбуков класса

тонкий-и-легкий качество в игровых AAA-проектах. Ключевые элементы для достижения заявленных результатов — ядро Zen, дающее скачок производительности более 50% относительно мобильных чипов Cag-rizo, и графические ядра Vega, увеличившие производительность GPU на 40% против текущего поколения. А по энергоэффективности Ryzen Mobile лучше вдвое. Ryzen Mobile изготовлен по техпроцессу 14 нм вместо 28 нм. Для бизнес-сегмента рынка компания готовит процессор Ryzen Pro, и, по словам руководства AMD, все 5 основных производителей ПК уже имеют планы использования Ryzen в своих компьютерах. Информацию о производительности и цене



» По результатам эталонных тестов новый процессор AMD Ryzen Pro обходит своего прямого конкурента Intel Core i5.

новых Ryzen Pro AMD не раскрывает, пока лишь заявив, что они «будут выглядеть достойно» рядом с продуктами Intel.

ЗНАЙ НАШИХ

В «высшей лиге» проектирования

Российская САПР поддерживает технологии виртуальной реальности.

Лаборатория виртуальной реальности российской компании «Топ Системы» завершила разработку интегрированного модуля, обеспечивающего использование технологий виртуальной реальности (VR) при работе с САПР T-FLEX CAD. Это первое в России и одно из первых в мире применений технологии виртуальной реальности непосредственно в системе проектирования.

Разработчики компании «Топ Системы» создали дополнительный модуль T-FLEX CAD VR, позволяющий работать с 3D-изображением в виртуальном пространстве при помощи шлема виртуальной реальности HTC Vive. Модель в виртуальном пространстве изменяется мгновенно при любых изменениях модели — добавлении, удалении перемещении объектов сцены, проигрывании сценариев анимации сборки/разборки и т.д. Навигация внутри виртуального пространства осуществляется при помощи штатных инструментов

управления VR-шлемом и обеспечивает перемещение в пространстве и изменение точки взгляда наблюдателя. При помощи специального манипулятора пользователь может выбрать требуемый ему объект сцены для осуществления редактирования.

Реализация такого модуля стала возможной благодаря широким возможностям, предоставляемым новой графической библиотекой, построенной на стандарте OpenGL. Эта библиотека лежит в основе графического интерфейса системы T-FLEX CAD, и является собственной разработкой компании «Топ Системы». Сведения о поддержке в T-FLEX CAD технологии VR привели к тому, что данной функциональностью заинтересовались пользователи систем T-FLEX в Германии и Чехии.

Технология T-FLEX CAD VR может быть использована для решения самых разных задач: виртуального обучения, создания презентаций готовой продукции и видеоруководств, улучшения эргономики будущих



➤ Возможности технологии T-FLEX CAD VR были продемонстрированы 30 мая в Москве, на ежегодно проводимом компанией «Топ Системы» IT-форуме.

изделий, контроль качества и добавлению в проект звуковых и текстовых аннотаций находясь непосредственно в виртуальном пространстве модели.

НАКОПИТЕЛИ

Места нет? Место есть!..

С помощью промышленного SAS SSD компания WD стремится удовлетворить растущие потребности в объемах для хранения данных.

Western Digital Corp. объявила о выпуске ориентированного на дата-центры высокоэффективного SAS SSD форм-фактора 2,5", который поможет компаниям справиться с растущими потребностями виртуализированных систем хранения, онлайн-обработки транзакций, аналитики баз данных, частных и гибридных облаков. WD Ultrastar SS300 разработан совместно с Intel, использует интерфейс SAS 12 Гб/с, обеспечивает последовательные скорости чтения-записи 2,1 Гб/с и 2,05 Гб/с соответственно, и IOPS произвольного чтения-записи до 400 тыс. и 200 тыс. операций соответственно.

Привод поставляется с памятью MLC NAND-флеш емкостью от 400 ГБ до 3,2 ТБ, или TLC NAND-флеш емкостью от 480 ГБ до 7,68 ТБ. MLC NAND сохраняет 2 бита, а TLC NAND — 3 бита на транзистор, и нуждается в более сложном ПО для контроля

равномерности износа и коррекции ошибок, но обеспечивает большую плотность. В зависимости от требований к питанию, пользователи могут выбрать SSD на 9, 11 или 14 Вт. Накопитель также использует технологию Intel 3D NAND: ячейки уложены стеками друг над другом, что позволяет значительно увеличить плотность и резко снизить издержки производства.

Ultrastar SS300 поставляется в четырех вариантах эксплуатационного ресурса, которые варьируются от 0,5 до 10 DWPD; AFR (годовая интенсивность отказов) составляет 0,35%, MTBF (среднее время между отказами) — 2,5 млн часов. Так, версия Ultrastar SS300 3,2 ТБ с параметром 10 DWPD способна выдержать до 59 ПБ (петабайт) произвольных записей за весь срок эксплуатации, что эквивалентно записи 16 ТБ данных в день в течение пяти лет.

Ultrastar SS300 предлагает разнообразные технологии для защиты данных, в т.ч.

Instant Secure Erase и Self-Encrypting Drive, которые совместимы со спецификацией TCG Enterprise и сертификатом FIPS 140-2. Поставляется новый диск исключительно производителям оборудования (по этой причине его цену Western Digital Corp. не называет).

➤ **Грегори Вонг (Gregory Wong), ведущий аналитик Forward Insights:** «WD Ultrastar SS300 дает оптимальный баланс производительности, емкости и цены для самых требовательных рабочих нагрузок».



НА ПОРОГЕ БУДУЩЕГО

Квантовый прорыв IBM

«Самый быстрый» на сегодня 17-кубитный квантовый процессор станет основой первых коммерческих систем в рамках программы IBM Q.

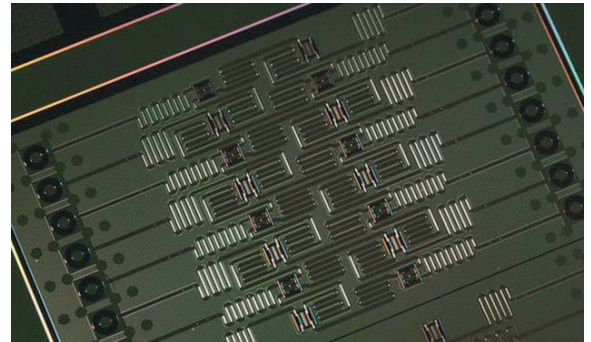
В IG BLUE резко продвинулась по пути развития квантовых компьютеров; созданы и протестированы 2 новых квантовых процессора, значительно превосходящих предыдущую 5-кубитную модель: 16-кубитный квантовый процессор, к которому уже предоставлен свободный доступ через облачный сервис IBM Cloud, и 17-кубитный прототип для коммерческих систем, «эффективнее предшественника, по крайней мере, в 2 раза» (для оценки производительности квантовых вычислений IBM ввела характеристику «квантовый объем [Quantum Volume]», учитывающую количество и связность кубитов, а также качество и параллелизм квантовых вычислений). 17-кубитный процессор станет ядром первых систем IBM Q — начатого в марте 2017 г. проекта создания коммерчески доступных универсальных квантовых вычислительных систем для деловых и научных приложений.

IBM Cloud уже более года выдает публичный доступ к квантовым процессорам

компании IBM; за это время исследователи и энтузиасты провели на них более 300 тыс. экспериментов.

«Представленные сегодня значительные технические усовершенствования позволят IBM масштабировать будущие процессоры до 50 или даже более кубит, достигнув вычислительных мощностей, лежащих далеко за пределами возможностей современных классических компьютерных систем», утверждает Арвинд Кришна [Arvind Krishna], руководитель подразделения IBM Research and Hybrid Cloud.

В отличие от классических битов, состоящих из единиц и нулей, квантовые компьютеры оперируют кубитами (квантовый разряд, от quantum bit), которые могут одновременно находиться в состоянии и «1», и «0» — ситуация, известная как суперпозиция. Другое ключевое понятие квантовой физики — запутанность: явление, при котором квантовые состояния двух или большего числа объектов оказываются взаимозависимыми. Именно суперпозиция



➤ Нарастание вычислительных возможностей квантовых систем открывает исследователям доступ к новым рубежам.

и запутанность ответственны за большую часть получаемой дополнительной вычислительной мощности: возникающий квантовый параллелизм, при котором операции производятся одновременно и с «0», и с «1», позволяет задействовать в квантовых компьютерах принципиально новые алгоритмы вычислений.

СДЕЛАНО В РОССИИ

Высокомерен ли Эльбрус?

«Росэлектроника» представила на «ЦИПР» первые компьютеры на базе микропроцессора «Эльбрус 8С».

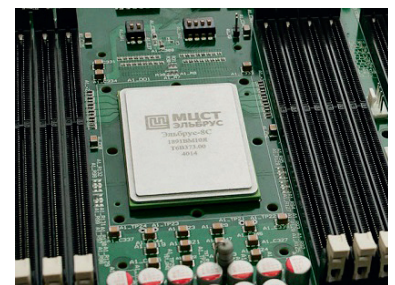
Н а состоявшейся 24–26 мая в Иннополисе (республика Татарстан) конференции «ЦИПР 2017» Объединенный холдинг «Росэлектроника» (входит в Госкорпорацию Ростех) представил первые образцы персональных компьютеров и серверов на базе микропроцессоров «Эльбрус-8С». Новая техника имеет повышенную производительность и гарантирует пользователям высокий уровень защиты информации. Восьмиядерный чип производится по технологии 28 нм. По сравнению с «Эльбрус-4С», пиковая производительность нового процессора выше в 3–5 раз, пропускная способность каналов ввода-вывода — в 8 раз.

«Это новое поколение отечественной вычислительной техники. Все этапы сборки осуществляются на наших производственных площадках или на предприятиях

отечественных партнеров. Все это гарантирует высокий уровень информационной безопасности оборудования», — комментирует заместитель генерального директора АО «Росэлектроника» Арсений Брыкин. — Мы ожидаем, что первая опытная партия персональных компьютеров на основе нового процессора будет готова уже к концу 2 квартала 2017 г.»

Установочная партия 2- и 4-процессорных серверов на основе «Эльбрус-8С» будет выпущена к концу 2017 г. Новые серверы предназначены для обработки больших объемов информации, в т. ч. в режиме реального времени. Базовой ОС для «Эльбрус 801-PC» и серверов является ОС «Эльбрус». Она построена на основе ядра Linux и поддерживает множество приложений с открытым исходным кодом. Для платформы Эльбрус-8С предусматривается система

➤ Эльбрус-8С производится по техпроцессу 28 нм, имеет 8 ядер (1,3 ГГц), кэш L2 4 МБ, кэш L3 16 МБ, 4 контроллера памяти (DDR3-1600); пик производительности — 250 Гфлоп.



двоичной совместимости с бинарными кодами x86/x86-64. Также обеспечена возможность разработки прикладного ПО, тесты для самодиагностики аппаратуры.

В составе объединенной «Росэлектроники» разработку и внедрение программно-аппаратных платформ «Эльбрус» ведет Институт электронных управляющих машин (ИНЭУМ) им. И. С. Брука.

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

НА РАДОСТЬ ГЕЙМЕРАМ

Поступь Legion'ов

Ноутбук Lenovo Legion Y920, преемник Ideapad Y900: серьезные возможности, серьезная цена.

Lenovo представила 17-дюймовый ноутбук Legion Y920 с CPU Core i7-7700HQ или Core i7-7820HK; высочайшее качество отображения игрового процесса обеспечивают GPU GeForce GTX 1070 и матрица IPS 75 Гц 1080p на технологии Nvidia G-Sync. Y920 комплектуется 16 ГБ ОЗУ DDR4, различными вариантами накопителей, включая 512 ГБ SSD плюс 1 ТБ HDD, портами Gigabit Ethernet, полномерным HDMI, DisplayPort и Thunderbolt, четырьмя портами USB 3.0 Type-A, кардридером SD; механической клавиатурой с многоцветной RGB-подсветкой и добавочным 10-клавишным блоком; взаимодействует со шлемами VR. Модель с CPU Core i7-7820K дополнена кнопкой Turbo, по нажатию на которую прилагаемая утилита Turbo Boost выполняет безопасный



► **Мощный ноутбук для профессиональных геймеров Legion Y920 стал дебютным ноутбуком Lenovo со смешанной механической клавиатурой.**

разгон процессора. При весе 4,58 кг Y920 не очень переносим, но для игрового ноутбука важнее эффективность охлаждения и уровень шума от ее работы; тесты pc-world.com показали, что у Y920 они вполне приемлемы. Legion Y920 стоит \$2700. Прямой конкурент Legion Y920 — Dell Alienware 17 R4, но его цена — \$2400, а на разницу в \$300 докупается немало игр.

С ЭФФЕКТОМ ПРИСУТСТВИЯ

Смотри на вещи шире!

35-дюймовый монитор Acer Predator Z35P оснащен матрицей VA, покрывающей 100% пространства sRGB.

Ультраширокий изогнутый монитор Acer Predator Z35P уже доступен для предварительного заказа на Amazon по цене \$1100, с отправкой в последующие три недели. Новинка комплектуется 35-дюймовой матрицей VA с разрешением 3440×1440 пикс и соотношением сторон 21:9, имеет время отклика 4 мс, яркость 300 нит, контрастность 2500:1, частоту обновления 100 Гц, по одному порту HDMI 1.4 и DisplayPort 1.2, четыре порта USB 3.0, два встроенных динамика на 9 Вт и подставку, обеспечивающую углы наклона от -4 до 35 градусов.



► **Именно вогнутый, как у нового Acer Predator Z35P, экран лучше всего подходит для игр за счет создания эффекта полного погружения в происходящее.**

вокруг быстро движущихся на экране объектов; собственная технология Acer Vision-Care обеспечивает дополнительную защиту глазам пользователя.

По информации TFT Central, в меню Z35P есть встроенная функция, позволяющая увеличивать частоту обновления до 120 Гц — что вполне вероятно, поскольку предшественник, Z35 с разрешением 2560×1080 пикс, предоставлял возможность разгона до 200 Гц.

В июле Acer обещает представить еще один свой сногшибательный монитор 4K HDR G-Sync, XB272-HDR. **LXF**

Монитор ориентирован в первую очередь на геймеров. Помимо кривизны 1800R, создающей эффект присутствия (несколько таких мониторов создадут полный визуальный круг с радиусом 1800 мм), поддерживаются дополнительные технологии Nvidia: G-Sync, позволяющая синхронизировать скорость вывода кадров совместимой картой GeForce GTX с частотой обновления монитора, и Nvidia Ultra Low Motion Blur, уменьшающая размытость и ореолы

Новости короткой строкой

► Если обратиться к файлу на системном диске Windows 7, 8 и 8.1, используя как имя каталога комбинацию \$MFT (например, C:\\$MFT\<любое значение>), система подвисает, и не помогает даже перезагрузка. Источник: www.cnews.ru

► Иранские разработчики закрыли проект Parsix GNU/Linux, основанный на пакетной базе Debian пользовательский дистрибутив Linux. Источник: www.parsix.org

► Компания АстроСофт выпустила тестовую версию новой российской ОС реального времени МАКС с открытым кодом для IoT и встроенных систем. Источник: www.astrosoft.ru

► loadlibrary от Тэвиса Орманди [Tavis Ormandy] из Google Project Zero позволяет загружать Windows DLL в Linux. Источник: www.theregister.co.uk

► Bash-скрипт GNOME Layout Manager позволяет пользователям Gnome настроить свой рабочий стол в стиле Unity, Windows 10 или MacOS. Источник: github.com

► Intel намерена интегрировать поддержку Thunderbolt 3 в свои будущие процессоры и сделать спецификацию Thunderbolt 3 доступной бесплатно. Источник: newsroom.intel.com

► Госкорпорация Ростех, компании OMMG Technology и «Майнд Лабс» создают отечественный аналог Skype с высоким уровнем безопасности. Источник: rostec.ru

► 25 мая вышел первый стабильный релиз Devuan Jessie 1.0 (ветка Debian без systemd), а на 17 июня запланирован выход Debian 9.0 Stretch. Источники: devuan.org и lists.debian.org

► Специалисты НЦИ создали отечественную ОС «ОСь» и решения на ее базе по автоматизации рабочих мест и информационным системам. Источник: rostec.ru

► Первым автомобилем с ИИС на базе свободной платформы Automotive Grade Linux (AGL) станет новая Toyota Camry XV70 поколения 2018 г. Источник: www.linuxfoundation.org

► Линейку процессоров Intel Core i9 Skylake-X возглавила 18-ядерная модель Core i9-7980XE. Источник: www.intel.ru

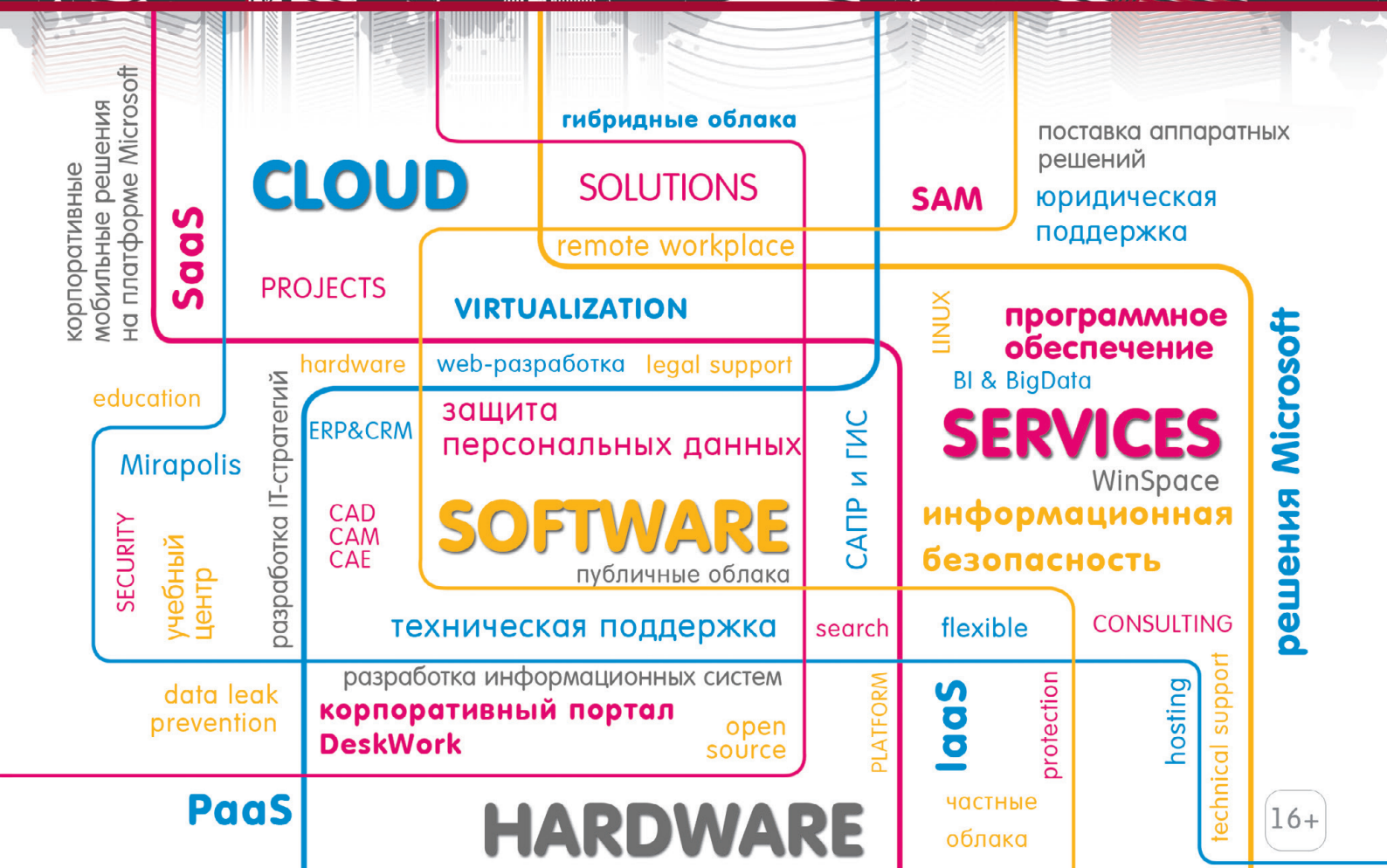
softline®

Cloud Software Hardware Services

20+

Years in IT

IT-архитектура вашего бизнеса



8 (800) 232-00-23

+7 (495) 232-00-23

www.softline.ru

info@softline.ru



ГНУ/Линуксцентр

*Ваш поставщик свободного программного
и аппаратного обеспечения*

**Комплекты
легализации СПО**

**Дистрибутивы
GNU/Linux и СПО**
на DVD и загрузочных
флэшках

**Дистрибутивы
GNU/Linux
и СПО**
с сертификатами
ФСТЭК, ФСБ
и Минобороны

**Межсетевые
экраны**
с сертификатами
ФСТЭК, ФСБ
и Минобороны



**Свободное
аппаратное
обеспечение**

Arduino, oLinuxino,
Cubieboard, Raspberry Pi,
Intel Edison, Digilent,
3D-принтеры
и робототехнические
конструкторы

**Аппаратное
обеспечение
с прошивками
на базе СПО**

**Обучающая
литература**

Атрибутика

Фирменный магазин и сервис-центр

Санкт-Петербург, пр. Медиков, 5, корп. 7

+7 812 309 06 86 | www.linuxcenter.ru





Новинки программного и аппаратного обеспечения в описании наших экспертов



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

Косметика или оптимизация?

Стали доброй традицией выходы очередной версии Cinnamon дважды в год — весной и осенью. Нынешняя весна ознаменована релизом за номером 3.4, хотя никакой информации о нем в предшествующие полгода на сайте разработчиков не было. И потому можно было не ждать особых новшеств; ну разве что косметических.

На первый взгляд так оно и было: не считать же важной вехой desktop-строения возможность не только вертикального, но и горизонтального размещения иконок на рабочем столе. Однако существенные изменения притаились в «нутре» среды: демон настройки *cinnamon-settings-daemon*, ранее запускавшийся как единый процесс, был разбит на составные части, согласно отдельным модулям Центра управления (управление питанием, настройка экрана и устройств ввода, и т.д.). Каждая из них запускается как отдельный процесс, не затрагивая остальные части десктопа. Настройка файлового менеджера *Neto* тоже разделена на два изолированных процесса: один отвечает за окна файлового менеджера, а другой — за конфигурирование рабочего стола. Все это должно повысить устойчивость среды и понизить потребность ее в ресурсах. И хотя по цифрам это не очень заметно, субъективно отзывчивость среды возросла. Не обошлось, конечно, и без косметики, более всего приложенной к настройкам экранной заставки. Но это — мелочи. Отмечу лишь, что тенденция развития среды Cinnamon — обходиться без революций и потрясений — продолжается. alv@posix.ru

Сегодня мы рассматриваем:

Parrot Security OS 3.5 14
Этот дистрибутив со специализацией по безопасности осмелился бросить вызов сопернику, находящемуся под покровительством грозной богини Кали! И в общем не подкачал.

4MLinux 21.0 15
По словам автора, это единственный Linux-дистрибутив, способный автоматически самовосстанавливаться. Причем его легко поддерживать в актуальном состоянии всего одной командой.

FreeNAS Corral 16
В популярном решении для сетевого хранилища данных переработали

графический интерфейс, что позволяет творить из него всевозможные чудеса — например, шифровать томъ ZFS.

Google WiFi 17
Одержимые Интернетом Вещей сотрудники Google разработали ячеистый роутер, способный охватить самые отдаленные уголки вашего дома, включая подвал.

Civilization VI 18
Как и всегда в игре *Civ*, мы строим империю, ведем дипломатию и отражаем воинствующих полководцев. Но в как никогда реалистичной графике и увлекательной живой истории.



➤ Роутер, который можно засунуть в карман.



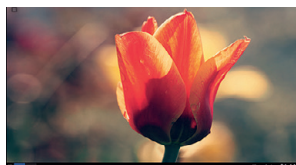
➤ Под крылышком у попугая: Parrot предлагает лучший инструментарий цифровой криминалистики.



➤ Прекрасно реализованный способ заставить вас принимать сложные решения на каждом повороте.

Сравнение: Дистрибутивы KDE с. 22

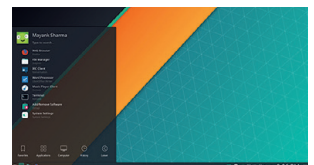
Chakra



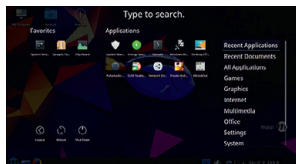
KaOS



Manjaro



Maui



Netrunner



Если вы любите обустроить свою систему во всех ее аспектах, рабочий стол KDE всегда готов вам помочь. Рассматриваем дистрибутивы, официально предлагающие KDE по умолчанию.

Parrot Security 3.5

Современный сознательный Робин Гуд, Шашанк Шарма, тестирует меры безопасности, сохраняя анонимность — пока что...

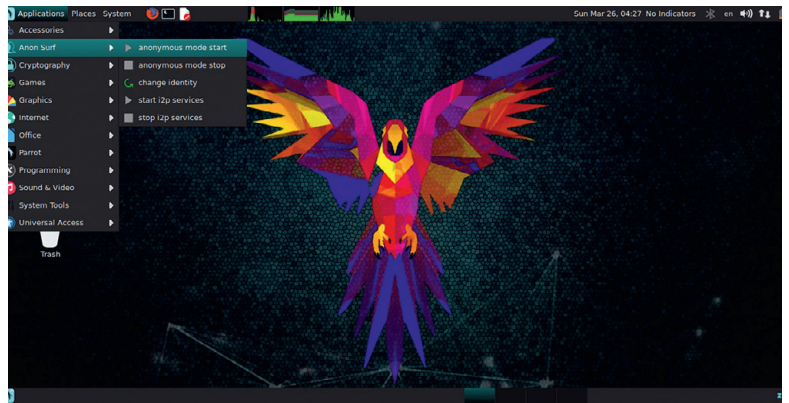
Вкратце

» Дистрибутив базе Debian с непрерывным циклом обновления, доступный как устанавливаемый live-носитель. Помимо широкого набора популярных инструментов для тестирования на вторжения, в нем также множество инструментов для обеспечения приватности и шифрования. Хотя сравнение с Kali Linux кажется естественным, дистрибутив может также применяться в качестве обычного настольного, для тех, кто заботится о конфиденциальности и безопасности.

Оценка надежности ПО, в сущности, состоит в том, чтобы защитить инфраструктуру вашей сети от действий злоумышленников. И уже довольно давно существуют специализированные дистрибутивы с огромным набором популярных инструментов, призванные вам в этом помочь, и многие из них борются за лидерство в сфере оценки уязвимости.

Одним из таких является ОС Parrot Security, на базе Debian, с непрерывным циклом обновления. Хотя его структура идентична дистрибутиву Kali Linux, самому, пожалуй, популярному для проверки безопасности, у Parrot Security всё же есть пара тузов в рукаве, чтобы произвести впечатление на новичков и опытных администраторов. Во-первых, в отличие от своих многочисленных собратьев, работающих в режиме live-окружения, Parrot можно установить на диск, а значит, в нем немало приложений для работы, и можно установить еще больше, используя репозитории. Все специализированные инструменты размещены в меню Parrot, удобно разделенном на категории и, при необходимости, подкатегории. Например, меню Information Gather [Сбор информации] далее делится на анализ SSL, анализ DNS и т. д.; но есть и универсальные инструменты, такие как *ntop*, которые не отнеси ни к какой подкатегории. Меню Wireless Testing [Тестирование беспроводной сети] также включает подменю, в том числе для инструментов 802.11 и Bluetooth.

Parrot разрабатывали в сотрудничестве с Caine, и в нем лучший инструментарий в области цифровой криминалистики, с наилучшими средствами анализа,



» Можете опробовать этот рабочий стол на базе Mate на соседском Wi-Fi. Только в целях обучения. Или розыгрыша. [Ред.: — НЕТ!]

предоставления доказательств и отчетности. Ради вашей безопасности дистрибутив включил несколько инструментов шифрования для защиты данных. В качестве бонуса, в нем также изначально имеется поддержка Tor, в частности, *torbrowser*, *torchat*, *Anonsurf*, а также другие инструменты конфиденциальности, чтобы замаскировать ваше присутствие в Интернете. В последнем релизе также есть поддержка гостевых виртуальных машин *VirtualBox* и *VMware*; у других аналогичных систем такого нет.

Детские болезни

Для любого дистрибутива Linux, особенно специализированного, установка является одной из ключевых задач. Пользователи Parrot могут выбирать установку на USB, постоянную или временную, или на диск, не требующий предварительной загрузки в среду live. Более того, помимо стандартного установщика на *Curses*, есть еще *GTK*-управляемый, для любителей работать с мышью. Последнее оказалось весьма полезным, потому что дистрибутив отказывался запускать установщик из live-окружения, выдавая ошибки. Кроме того, установка не удалась во время одного из наших тестов, когда мы решили установить по отдельности */home* и */tmp* для создания разделов с использованием зашифрованного LVM. За исключением этих двух нестыковок, дистрибутив работал безупречно даже на машинах с небольшим объемом оперативной памяти.

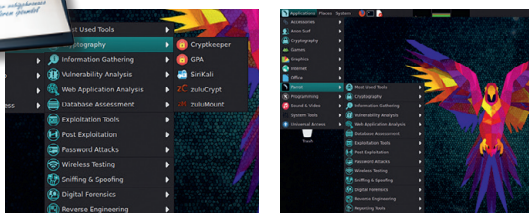
В чем дистрибутив отстает от лидера Kali Linux, так это в документации, предельно краткой, хоть и функциональной. До этого года у проекта даже отсутствовал

свой форум. Недавно запущенные форумы довольно активны, но поскольку постов там пока немного, полноценным информационным ресурсом это назвать трудно.

Основанную на Debian Parrot Security OS часто незаслуженно сравнивают с ее более успешным аналогом, Kali Linux, несмотря на то, что она больше похожа на Caine. Но репозиторий инструментов Parrot сравним с Kali, и дополнен целым арсеналом программ для криптографии, вкпе с различными инструментами анонимизации, чтобы, если вы того пожелаете, ваше присутствие в Интернете было всегда замаскировано, а коммуникации — зашифрованы. В общем, Parrot Security OS представляет собой подходящим решением для проведения тестов на вторжение с достаточным количеством инструментов охраны конфиденциальности, чтобы Сноуден им гордился. Как шепчет робин-гудовское в авторе... [LXF]



Свойства навскидку



Талантливый во всем

Всё, что нужно для предельной безопасности, шифрования и конфиденциальности в одном аккуратном дистрибутиве.

Удобные меню

Свои огромные возможности Parrot удалось вписать в хорошо продуманные, удобные для навигации меню.

LINUX FORMAT Вердикт

Parrot Security OS 3.5

Разработчик: Frozenbox Network
Сайт: <http://parrotsec.org>
Лицензия: Разные

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

» Полноценный дистрибутив с инструментами по трем категориям. Еще улучшить бы форум и документацию.

Рейтинг 7/10

4MLinux 21.0

Амбициозно минималистичные дистрибутивы могут быть весьма заманчивой приманкой, но клюнет ли на это **Шашанк Шарма**?

Вкратце

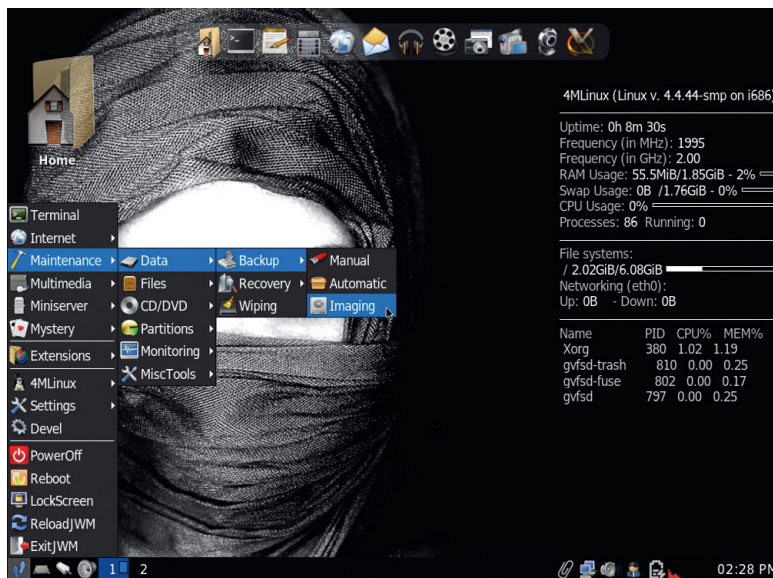
» Минималистичный дистрибутив со множеством уникальных особенностей. С ПО в его составе, дистрибутив Live может применяться для техобслуживания, развертывания мультимедиа, запуска различных серверов и даже чтобы предаться старым добрым играм. Пользователи также могут исследовать Antivirus Live CD, BakAndImgCD и SSS — это всё официальные ветки.

4М в названии обозначает четыре направления для этого в одиночку разработанного дистрибутива: Maintenance, Media, Mini-server и Mystery. Если вы решите установить его, то 500-МБ Live-дистрибутив требует менее 1 ГБ памяти. По части Maintenance, 4MLinux предлагает *Photorec*, *Testdisk* и различные инструменты, которые помогут восстановить данные из поврежденной установки Windows и управлять разделами. Также включены инструменты, помогающие проигрывать ассортимент аудио- и видеофайлов, со всеми соответствующими предустановленными кодеками. Также 4MLinux годится для запуска FTP-, SSH- и HTTP-серверов. Компонент Mystery ссылается на игры в комплекте, включая оригинальные *Quake* и *Doom*, *Tetris*, *Pacman*, *Sokoban* и другие.

Уникальный подход

4MLinux провозгласил себя единственным Linux-дистрибутивом, способным автоматически самовосстанавливаться. Это стало возможным благодаря выбору *Busybox* как службы инициализации. Помимо базовой системы, которая включает *Busybox*, *Bash* и файлы ядра, при первом запуске система инициализации строит различные каталоги, файлы конфигурации в */etc* и т.д. При каждой последующей загрузке 4MLinux перепроверяет каждый из этих компонентов и при необходимости восстанавливает.

Выбор *Busybox* также означает, что дистрибутив должен запускаться от имени суперпользователя-root. Но в целях безопасности дистрибутив автоматически создает системных пользователей, как, например, при запуске браузера.



» Благодаря *JWM* в качестве оконного менеджера, 4MLinux легко оправдывает свое название с таким пакетом ПО.

Если ваше оборудование не определяется должным образом, запустите команду *udev* для определения и настройки всех подключенных устройств, по какой-либо причине не настроенных при загрузке. Еще одна уникальная функция — возможность самостоятельного перезапуска командой *restart*. чтобы применить изменения в настройке файлов без перезагрузки.

В отличие от большинства других дистрибутивов, пакеты 4MLinux являются tar-архивами, сжатыми Xzip; их называют «аддонами» [add-on — *англ.* дополнение]. Все установленные пакеты перечислены в папке */var/4MLinux*. Дополнительные пакеты, такие как драйверы, можно загрузить с <http://bit.ly/4Mdrivers>. Устанавливаются аддоны через специальный менеджер пакетов *zk* командой *zk addon_name.tar.xz*. А *zk update* обновит установленный 4MLinux до последней стабильной версии.

Хотя 4MLinux не отправляет обновления к своей коллекции пакетов, как в большинстве других дистрибутивов Linux, иногда в начале месяца он отправляет доработанные версии в устойчивый канал, поэтому *zk update* будет поддерживать вашу систему в актуальном состоянии. Кроме того, пакеты *Snap* в меню *Extension* не доступны по умолчанию. При щелчке по ним загружается скрипт, устанавливающий пакет.

В прошлых версиях у 4MLinux случались проблемы с настройкой беспроводных

карт. И, несмотря на правильное распознавание, он не смог подключить беспроводную карту Intel на одной из тестовых машин. Справедливости ради, отмечу: эта конкретная карта досадила многим пользователям Linux в разных дистрибутивах.

Документация (спартанская) выложена на сайте — <http://4mlinux.com>. Документы по установке и рабочему столу доступны в блоге с 2013 г., но при необходимости обновляются с каждой новой версией. FAQ дает ответы на многие вопросы, и пользователи могут помещать запросы на форуме 4MLinux на LinuxQuestions.org — разработчик-одиночка очень активен. **LXF**



Свойства навскидку

Свои скрипты

4MLinux поставляется с инструментами пользователя, такими как *udev*, *restart*, *zk*, *connect*, *server* и др.

Самолечение

Вы можете удалить или импортировать каталог */etc* и тем не менее перезагрузиться в рабочую установку.

LINUX FORMAT Вердикт

4MLinux 21.0

Разработчик: Збигнев Конояцкий
[Zbigniew Konojacki]
Сайт: www.4mlinux.com
Лицензия: GPLv3

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

» Уникальный дизайн; хорошо подойдет тем, кто ищет дистрибутив для задач техобслуживания.

Рейтинг 9/10

FreeNAS Corral

Шашанк Шарма рассматривает популярное решение NAS, которое, как и сам рецензент, обретает новую жизнь.

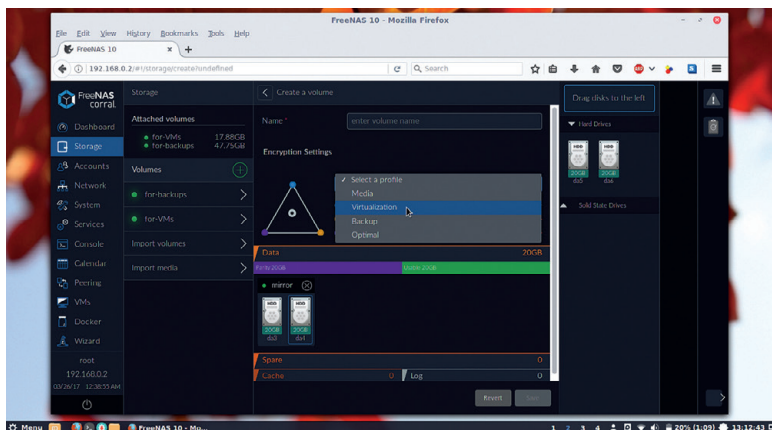
Вкратце

» Одно из самых популярных решений сетевых хранилищ с открытым исходным кодом, на базе ОС FreeBSD. Изобилует функциями, связанными с NAS, и может расширяться с помощью плагинов. Последний выпуск представляет собой важную веху для проекта. В открытых решениях NAS нет недостатка, но бесплатные NAS4Free, Amahi и Open Media Vault на базе Debian предназначены для продвинутых домашних пользователей.

Последний выпуск популярного решения сетевого хранилища данных — версия, переработанная снизу доверху. Названное при разработке FreeNAS 10, оно стало FreeNAS Corral, потому что теперь ОС может использоваться для сбора данных, хранения устройств, виртуальных машин и перемещаемых контейнеров под единым переработанным интерфейсом управления. Улучшенная поддержка для самовосстанавливающейся файловой системы OpenZFS с поддержкой копирования при записи [copy-on-write] гарантирует, что файлы никогда не будут перезаписаны; поддерживаются Active Directory и службы каталогов FreeIPA; и улучшены резервное копирование и функции репликации — вот часть «мелких» реформ. ZFS — одна из основных причин популярности FreeNAS, благодаря таким полезным функциям, как программный RAID, известный как RAID-Z, а также снимки файловой системы, которые могут создаваться по расписанию и храниться удаленно.

Настоящей изюминкой является добавление гипервизора FreeBSD, пригодного для создания и загрузки гостевых операционных систем изнутри вашего NAS. Если вы предпочитаете контейнеры, а не виртуальные машины, в этой версии также интегрирована поддержка для Docker.

Переработанный интерфейс управления через браузер теперь выглядит современнее и стал более интуитивно понятным. Наряду со всеми возможностями его предыдущего аватара, теперь можно будет познакомиться с новыми функциями виртуализации ПО. В FreeNAS Corral также входят



» Для ознакомления пользователей с новым интерфейсом разработчики FreeNAS создали подробные ролики, доступные на www.youtube.com/user/FreeNASTeam.

свыше десятка простых в использовании шаблонов. С их помощью вы получаете предустановленные и полностью настроенные конфигурации версий популярной ОС BSD и дистрибутивы Linux, такие как TrueOS, FreeBSD, CentOS, Linux Mint, Ubuntu и другие. Можно создать виртуальные машины из ваших собственных ISO-образов.

Под новым управлением

Графический интерфейс управления является одним из наиболее заметных изменений. Боковая панель уведомлений очень приятна, особенно когда вы начинаете продолжительный процесс, например, создание виртуальной машины из шаблона. Еще одна мощная функция — создание тома.

Для создания тома Corral предусматривает четыре готовых профиля, которые дают точный баланс производительности, мощности и избыточности. Например, при выборе Медиа в меню интерфейс выдаст максимальный весовой коэффициент безопасности и некоторую избыточность за счет производительности. Интерфейс позволяет менять веса и создавать графические профили пользователя — это мило.

А еще интерфейс позволяет устанавливать и использовать подключенные диски путем перетаскивания и сбрасывания их из пула доступных дисков в пул хранения — на языке ZFS, vdev. Эти виртуальные устройства vdev автоматически размещаются в форматы RAID в зависимости от числа дисков в пуле. Так, если вы добавили во vdev пять дисков, они будут размещены в RAID-Z3, а значит, вы не потеряете свои данные, даже если три диска в пуле выйдут

из строя. Для дополнительной безопасности тома ZFS можно зашифровать.

И кроме того, интерфейс позволяет добавлять диски в качестве запчастей, чтобы при сбое накопителя система автоматически заменила неисправный диск одним из пула. Аналогично, когда вы создаете общую папку, скажем, SMB или NFS, интерфейс позволяет указать все сопутствующие дополнительные опции внутри окна браузера. Для опытных участников FreeNAS в Corral есть новый интерфейс командной строки с поддержкой скриптов, и можно автоматизировать и контролировать каждый аспект. Если у вас стоит предыдущая версия, FreeNAS 9.10, то обновить ее до версии Corral можно без проблем. Разработчики FreeNAS планируют поддерживать версию FreeNAS 9.10 и выпускать исправления и обновления, «пока есть аудитория». LXF

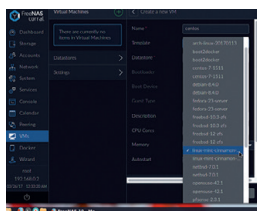


Свойства новскидку



Новый интерфейс

Предоставляет мощные функции в довольно интуитивной форме; доступен для пользователей различной степени мастерства.



Запуск VM

Для управления и питания виртуальной машины и контейнеров Docker используйте NAS, оснащенный Corral.

LINUX FORMAT Вердикт

FreeNAS Corral

Разработчик: iXSystems
Сайт: www.freenas.org
Лицензия: BSD

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

» Благодаря новому интерфейсу и функциям подходит и для дома, и для предприятий.

Рейтинг 9/10

Google WiFi

Зачем вам дома всего один роутер? **Джо Осборн** выбрал три mesh-роутера!

Спецификация

- » **Wi-Fi IEEE** 802.11a/b/g/n/ac, AC1200 2x2 Wave 2 Wi-Fi (расширенный mesh; двухдиапазонный 2,4 ГГц и 5 ГГц, формирование луча TX); готовый Bluetooth Smart
- » **CPU** 4-ядерный ARM 710 МГц
- » **ОЗУ** 512 МБ
- » **SSD** 4 ГБ, eMMC flash
- » **Направленность** Явная и неявная для диапазонов 2,4 и 5 ГГц
- » **Порты** 2xГБ Ethernet для каждой точки (1 WAN, 1 LAN)
- » **Габариты** 106,1x68,7 мм (диаметр x высота)
- » **Вес** 340 г

Роутеры и коммутаторы мертвы — будущее за Wi-Fi mesh- [ячеистыми] или трехполосными [tri-band] системами, использующими стандарт 802.11s. Естественно, что технология «умный дом», которой одержим Google, повсюду использует Google WiFi.

Как оказалось, Google может иметь отлично разработанную лучшую mesh-систему Wi-Fi на сегодняшний день. Компании удалось создать систему, которая предлагает больше mesh-устройств, чем конкуренты, с акцентом на простой установке и управлении. Что в результате? Мы никогда больше не захотим и взглянуть на свой шлюз.

Google WiFi стоит £ 229 за набор из двух устройств: одно — главная «WiFi Point [Точка Wi-Fi]» (ее вы подключите к модему или шлюзу) и второе — подчиненная WiFi Point; но при желании всегда можно добавить еще. Одиночное устройство Google WiFi можно приобрести за £ 129, и Google обещает, что три WiFi Points смогут охватить до 418 кв. м в (большом) доме. Так или иначе, Google предлагает больше устройств за меньшие деньги в сравнении с конкурентами: например, Netgear Orbi и другие будут стоить как минимум £ 320 за то же число.

Mesh означает, что «роутером» системы может работать любое из устройств, тогда как другие могут раздавать проводной интернет (рожденный беспроводным способом) с их встроенных Ethernet-портов точно так же, как и беспроводной. Все устройства заряжаются через USB-C.

Настройка столь же безупречна, как и дизайн оборудования Google; для упрощения процесса используются бесплатные приложения для iOS и Android. [Ред.: — Помните, что опция настройки НЕ привязана к конкретному браузеру, надо иметь пригодное устройство Android или iOS. Плохо, Google.]

Сканируем QR-код, далее приложение велит дать имя вашей сети и создать пароль, затем подключает дополнительные точки Wi-Fi, и снабжает их в приложении метками для ссылки. «Роутеру» требуются секунды на распознавание Wi-Fi Point, чтобы они начали трансляцию. Опытные пользователи, имейте в виду: вы не получите ту же глубину доступа, которую дает Netgear Orbi, так что для вас нет переключателя диапазона чистот.



» Это устройство работает так же просто, как выглядит.

Приложение предлагает массу полезных функций, таких как постоянный мониторинг вашей сети, ее Точек и подключенных устройств. В приложении есть встроенный тест скорости интернета, mesh-тест, которые замеряет рабочее состояние соединений у ваших точек, и Wi-Fi тест, который меряет прочность связей внутри сети. Также можно временно установить приоритет пропускной способности для одного устройства, следить за устройствами «умного дома» и приостанавливать доступ к Интернету для определенных устройств в семейном кругу — всё это внутри одного приложения.

Мы увидели, что результаты производительности системы Google WiFi впечатляют не меньше, если не больше, чем наблюдавшиеся у Netgear Orbi. Google WiFi выжал максимум из наших сервисов на 100 Мбит/с, на которые способен любой другой роутер, и сумел сделать это в любой комнате нашего, правду сказать, небольшого дома.

У нас получилось направить поток видео 4K от Netflix к нашему плееру Roku Premiere, который установлен в подвале,

не хуже, чем сыграть в *Overwatch* в офисе, где располагался модем: без вопросов.

Mesh-системы Wi-Fi типа Google WiFi акцентируют внимание не столько на пропускной способности, сколько на покрытии, но этот продукт доставляет ее независимо. Настоящее преимущество Google WiFi перед другими — это ширина его покрытия за свою цену. **LXF**

LINUX FORMAT Вердикт

Google WiFi

Разработчик: Google
Сайт: <https://madeby.google.com/wifi>
Цена: £ 229

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

» Не самый мощный или точно контролируемый, но самый простой и управляемый роутер из тех, что мы настраивали.

Рейтинг 9/10

Civilization VI

Окруженный детьми, словно голограмма их пра-пра-прадеда, **Ти Джей Хейффер** рассказывает им о том, как на месте этого космодрома когда-то были поля.

Спецификация

Минимальные:

- » **ОС** Ubuntu 16.04, SteamOS (64-битные)
- » **Процессор** Intel Core i3 530, AMD A8-3870
- » **ОЗУ** 6 ГБ
- » **На диске** 15 ГБ
- » **Графика** Nvidia GeForce 650, 2 ГБ VRAM
- » **НВ** Поддержки видеокарт Intel и AMD (официальной) нет

Civilization VI — квинтэссенция того, какой следует быть игре для цифровых платформ. Никогда еще в этой серии игровая доска — сам этот мир — столь не воодушевляла каждую возможность, каждый выбор. Как всегда в *Civ*, мы строим империи, конкурируем за условия победного договора и отражаем воинствующих полководцев вроде этого подлеца Петра Великого. Но кроме того мы играем за, вместе и против доски. Леса, пустыни и богатые ресурсами тундры определяют судьбу нашей цивилизации, предоставляя нам блага и обременяя нас труднопреодолимыми слабостями. Орды варваров разоряли наши фермы, но и стимулировали развитие нашей военной техники. Великолепная и сложная динамика новых карт *Civ VI* не оставляет сомнений в том, что эта легендарная серия увенчала нового короля.

Несмотря на то, что *Civ VI* — это, пожалуй, самый смелый шаг к новшествам в этой серии, игроков со стажем такие изменения оттолкнуть не должны. Вы по-прежнему строите города, развиваете ремесла, тренируете воинские части, воюете в пошаговом режиме и разводите дипломатию. По старой памяти, это так близко к прежним *Civ*, что при встрече с чем-то ранее неизвестным нам достаточно намек советчика внутри игры, чтобы выбрать правильный путь.

Правда, новых функций настолько много, что порой они даже утомляют. По глубине и разнообразию систем это напоминает



» Формирование отрядов стало более гибким.

Civ, в которую уже добавлено два или три расширения — от новых Districts [Районов], которые выполняют специальные задачи и превращают города в нередко беспорядочное и вместе с тем приятное место — до целого «древа» технологий, служащих для развития культуры и цивилизации и складывающихся в своего рода пасьянс, где из политических бонусов создается уникальное правительство.

Связующее звено для всего этого — карта; сама карта, ее города, железные рудники и городские площади выглядят реальнее, чем когда-либо. Невозделанные поля лежат бесплодными, а по уровню суеты

на улицах можно угадать, сколько жилых блоков занято в коммерческом районе. Надо признать, это великолепный способ заставить нас безотрывно следить за тем, что происходит.

Технологические деревья и экран взаимодействия лидера — это единственные части интерфейса пользователя, которые скрывают от взгляда ваши растущие города. Экран включает полностью анимированные 3D-изображения всех действующих лиц — от Монтезумы до этого долболама Петра Великого, который считает себя крутым, раз у него есть усы и бонус в науке, благодаря тундровым слоям — хотя это чушь, и наши бомбардировщики уже готовы достать его второй по величине город со времен Атомного века, чтобы стереть дурацкую ухмылку с его лица.

Всё отлично озвучено, и диалоги снова сделались доступны на родном языке, как в *Civ V*.

Вы ни секунды не ощущаете, что заполнили каждую клетку явно «правильным» районом или улучшением и можно успокоиться. Всегда надо сохранять осмотрительность. Постоянно приходится чем-то жертвовать: мы, например, отстали в культуре, поскольку отвели единственную клетку, пригодную для театральной площадки, под запуск ракеты, чтобы вырвать победу в науке. Это фантастический, прекрасно реализованный способ заставить вас принимать сложные решения на каждом новом повороте реки и сделать так, чтобы ни один

» **Неисследованные регионы имеют приятный «пергаментный» вид.**



из построенных вами городов не походил на другие.

То, что иногда приходится действовать методом проб и ошибок, отчасти может разочаровать в первых нескольких гонках космической эры. Когда всё в новинку, вы можете не осознавать, что втиснули университетский городок туда, где несколько веков спустя предстоит отстраивать жилые кварталы. Не помешало бы иметь какой-либо инструмент городского планирования, с макетами, на которых показано, куда что идет.

Еще одна причина, почему карта стала в *Civ VI* гораздо более значимой — это ее связь с древом технологий и древом культуры. У каждой технологии и культурного достижения есть мини-цель, которая вызовет момент «Эврика» и тут же окупит половину стоимости. Если основать город рядом с океаном, активизируется развитие мореплавания. Строительство трех промышленных районов с заводами подталкивает вас к идеям коммунизма. (Да здравствуют строители экономической политики!)

Хотя не всё столь стереотипно. Принципы ведения войны и дипломатии в *Civ VI* те же, что и раньше, но отточены до предела. Особенно понравилось то, что те-



» Здания больше не привязаны к центру города.

середину между принципом «один отряд на одну клетку» версии *V* и методом *Clash of Doomstacks* из *IV*. Отряды поддержки, такие как медики и Великие генералы, могут прикрепляться к той же клетке, что и регулярная боевая единица вроде копьеносцев. В середине и конце игры вы также получаете возможность комбинировать два боевых отряда в корпус, а позже добавить третий, создав армию, которая является более мощной версией того же соединения и занимает только одну клетку. Это добавляет новые уровни и тактики ведения войны, способной стать предсказуемой и повторяющейся, как в *Civ V*.

Civ вдыхает жизнь во всяческие конфликты и переговоры. Новая основная тема от Кристофера Тина [Christopher Tin], *Sogno di Volare*, просто нереальна, привязчива и прекрасна, как и *Baba Yetu*. Правда, настоящее волшебство происходит за экраном меню, где у каждой цивилизации есть основная тема, которая становится всё сложнее и эпичнее по мере вашего путешествия сквозь века.

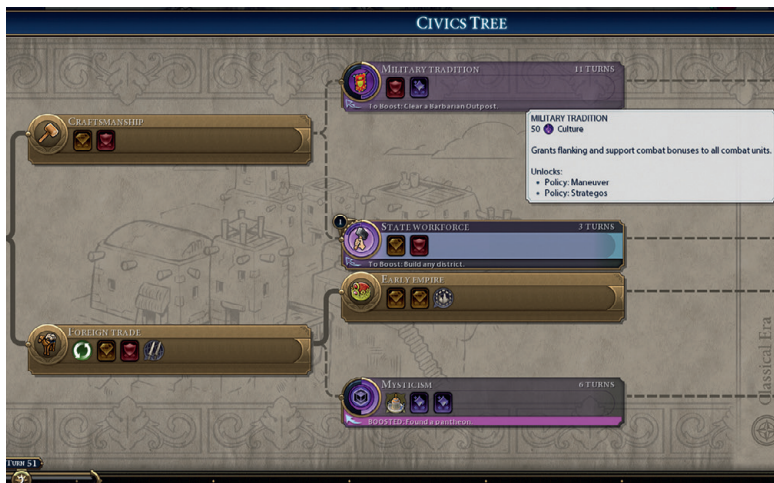
Оглядываясь на отстроенное нами, когда наши колонисты на Марсе практически

вырвали победу у Петра и его, естественно, усатых дружков, мы ощущали в каждой клетке дыхание истории. В кнах разросшегося делейско-калькuttского мегаполиса отражались воспоминания. Как мы провернули аферу при покупке клетки, чтобы получить доступ к углю. Как мы рыли бульдозером 3000-летний участок сельхозугодий, чтобы выбить место под это чудо промышленной эпохи. И как прямо рядом с местом, где осел наш первопоселенец, у подножия вершин, в течение нескольких поколений защищавших нас от мародерствующих армий, разросся новый лес, который мы посадили на месте бывшей лесопилки, чтобы объединить зеленую зону для Национального парка. Мы могли бы сказать вам, почему мы расположили каждую долину, степь и оазис именно так, и гораздо полнее, чем просто «Потому что холмы — перспективное место для шахт». Поскольку нашу империю формировала доска, и мы ее формировали, история цивилизации и наши решения накапливались и сопровождали нас от начала и до самого конца. И не что-либо, а именно это делает для нас данную *Civ* лучшей из возможных. **LXF**

Связующее звено — карта. Она реалистичнее, чем когда-либо.

перь руководителям ИИ даются задания (одно — открытое, а другое надо раскрыть посредством шпионажа, налаживания связей или наблюдения). Это проливает свет на предпочтения лидеров, теоретически позволяя оставаться у всех на хорошем счету в течение всей игры, если вы хотите угодить всем.

Если боевые действия действительно вспыхивают, то *Civ VI* выбирает золотую



» Добавилось древо культурного прогресса.

LINUX **Вердикт**
FORMAT

Civilisation VI

Разработчик: Aspyr
 Сайт: www.aspyr.com
 Цена: £ 50

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

» Пусть ИИ и сомнителен, вид, звук и системы делают *Civ VI* самой живой и увлекательной из всех стратегий 4X на этой зеленой земле.

Рейтинг 8/10



БУДЬТЕ ЗДОРОВЫ!

Колыбельная от хромбука

Вслед за Apple и Microsoft, компания Google добавила в свою ОС для ПК функцию ночной подсветки.

Поклонники Chrome OS узнали через социальный сайт новостей Reddit о предстоящей реализации в этой ОС функции «ночной режим», позволяющей изменить цветовой тон дисплея хромбука в сторону более теплых оттенков. Функцию вчерне назвали «Ночной свет [Night Light]», она доступна пока только в Canary, экспериментальной сборке Chrome OS; для ее активации надо щелкнуть сначала в правый угол полки [Shelf] Chrome OS, а в открывшейся панели настроек — по значку Night Light рядом с кнопкой управления питанием. Добавление функции ночного режима в ОС для ПК — тренд наших дней; ранее пользователям приходилось полагаться на приложения сторонних разработчиков (f.lux и подобные). В начале 2017 г. Apple добавила в свои ПК

Mac функцию ночного режима Night Shift (впервые — в iOS 9.3). В Creators Update for Windows 10 появилась аналогичная функция, тоже Night Light, переключаемая вручную или автоматически в заданное время, а пользователь может настраивать цветовую температуру по своему вкусу. Подсветка в ночном режиме снижает количество синего света, влияющего на пользователей-полуночников, а это положительно сказывается на процессе засыпания и самом сне. Синий свет подавляет выработку мелатонина, ключевого гормона для цикла сон-бодрствование, известного также как циркадный ритм. Большинство специалистов все же рекомендуют отключать ПК за час-два до сна. На выставке-конференции Macworld Expo в марте 2016 г. говорили, что мелатонин — не вся



► Регулировка цветовой температуры дисплея уменьшает излучение синего спектра в сторону более теплых оттенков.

проблема: действия, поддерживающие активность мозга, могут способствовать нарушениям сна. (Очередную битву в *World of Tanks* всяко лучше перенести на завтра).

НАРАЩИВАЯ ВОЗМОЖНОСТИ

Чипы для ИИ и обучения машин

CPU и GPU от ARM позволят мобильным устройствам работать с приложениями искусственного интеллекта и VR без обращения к облачным сервисам.

Накануне технологической выставки-конференции Computex 2017, прошедшей с 30 мая по 3 июня в Тайбэе, британский чипмейкер ARM, принадлежащий японской телекоммуникационной компании SoftBank, представил самые мощные на сегодня мобильные процессоры Cortex-A75 и A55, на микроархитектуре ARM DynamIQ. Возможности этой архитектуры позволяют конечным производителям (Qualcomm, MediaTek и т.д.) создавать чипы не с двумя кластерами разноплановых ядер, как ранее, а с единым кластером, состоящим из любого количества высокопроизводительных или высокоэффективных ядер.

Предназначенная для смартфонов уровня high-end флагманская модель Cortex-A75, по сравнению с предшественником

Cortex-A73, обеспечивает улучшение производительности при обработке однопоточных приложений на 22%, и до 50% при выполнении многопоточных задач. A75 рассчитан на использование в энергоемких SoC с возможностью потребления до 2 Вт, за счет этого производительность может быть повышена еще на 30% в устройствах с большими экранами. Согласно инсайдерской информации *The Inquirer*, на ядрах Cortex-A75 будет работать не представленный пока Qualcomm Snapdragon 845, которым Samsung планирует оснастить грядущий Galaxy S9. Cortex-A55, который производитель характеризует как «наиболее энергоэффективный CPU среднего уровня из когда-либо выпущенных ARM», обеспечивает на 15% большую, по сравнению со своим предшественником

► Высокопроизводительный Cortex-A75 обеспечивает производительность ноутбука при энергопотреблении смартфона; энергоэффективный Cortex-A55 займет место в не требующих высокой производительности мобильных устройствах, а также устройствах IoT.



A53, энергоэффективность, в 10 раз лучшую масштабируемость и вдвое большую производительность памяти.

Еще одной представленной ARM новинкой стал GPU Mali-G72, специально разработанный с расчетом на VR и предназначенный для использования в ориентированных на AI и машинное обучение SoC вместе с CPU A75 и A55. По сравнению с Mali-G71, G72 демонстрирует увеличенную в 1,4 раза производительность и повышение эффективности на 25%. ARM предполагает, что первые реализации новых технологий CPU и GPU появятся в Q4 2017 г. или Q1 2018 г.

ТРАНСФОРМЕРЫ

Киллер для Surface Pro 4

По многим параметрам HP Spectre x2 превосходит стареющий планшет Microsoft.

Трудно сказать, как будет выглядеть HP Spectre x2 второго поколения рядом с Surface Pro 5, если Microsoft когда-либо выпустит таковой, но по сравнению с Surface Pro 4 новый гибридный планшет HP, в котором исправлены многие огрехи первой версии, обладает гораздо лучшим соотношением функциональности и цены.

HP отказалась от процессоров Core M в пользу 7-го поколения CPU Intel Kaby Lake; графика Iris Plus, благодаря 64 МБ встроенной DRAM, обеспечивает заметное преимущество по сравнению с HD-устройствами; устройство комплектуется ОЗУ LPDDR3 до 16 ГБ и SSD PCIe до 1 ТБ. Серьезно переработан экран, получивший IPS-матрицу с диагональю 12,3 дюйма, разрешением 3000×2000 пикс и максимальной яркостью 450 нит.

Первый Spectre x2 работал с Wacom-подобным пером, теперь замененным на технологию N-trig, используемую Microsoft

в своем Surface Pen (Microsoft даже купила N-trig).

Основное преимущество N-trig — емкостной сенсорный слой для восприятия пера, в то время как устройствам Wacom необходим увеличивающий толщину экрана отдельный дигитайзер. У первого Spectre x2 отсутствовала ИК-камера, что не позволяло задействовать функцию аутентификации Windows Hello; у нового Spectre x2 отсутствует ИК-камера в наряде с задней камерой Intel RealSense есть фронтальная ИК-камера. В оригинальном Spectre x2 подставка вытаскивалась неудобным слайдовым переключателем, в новом она просто вытаскивается вручную. HP увеличила силу удерживающих съемную клавиатуру магнитов, что улучшило “lapability” планшета (возможность устройства принимать различные положения).

Новый Spectre x2 оснащается двумя портами USB-C, однако без Thunderbolt 3 и со скоростью 5 Гб/с (половина от возможных у USB 3.1 10 Гб/с), но с поддержкой



» «Осязаемые» козыри обновленного гибрида HP Spectre x2: алюминиевый корпус, настраиваемая подставка с поддержкой углов до 165 градусов, акустика Bang & Olufsen.

DisplayPort и зарядкой USB-PD. Удивляет отсутствие USB Type A, однако HP предоставляет в комплекте решающий эту проблему брелок.

В продажу обновленный Spectre x2 поступит уже в июне; цена модели с Core i5, 8 ГБ RAM и SSD 128 ГБ составит \$999, версии с Core i7, графикой Iris Plus, 8 ГБ RAM и SSD 256 ГБ — \$1280.

АВТОМОБИЛИ БУДУЩЕГО

Вези меня, Android!..

В следующем поколении автомобилей Volvo появится полноценная Android OS.

В мае Google объявила о сотрудничестве с Volvo по разработке нового продукта на базе Android. По словам продукт-менеджера Android Хариса Рамича [Haris Ramic], новинка «позволит водителю управлять кондиционером, люком в крыше и окнами, искать объекты на Google Maps, слушать потоковое аудио от Spotify либо NPR или обратиться за помощью к Google Assistant — и для этого не нужен будет телефон». Новые системы представили на конференции Google I/O в Маунтин-Вью (Калифорния).

Спустя 3 года с момента своего выпуска, Android Auto поддерживается 300 моделями автомобилей и так называемых афтермаркет-стереосистем. Google выводит Android Auto за рамки отдельного приложения, предоставляющего водителям лишь средства навигации, потоковое аудио и сервисы для коммуникации. Реализация в информационно-развлекательной системе автомобиля всех возможностей ОС Android

выгодна Google в преддверии масштабного распространения «подключенных устройств [Connected Devices]». Согласно недавнему отчету Cisco, к 2021 г. число подключений M2M (включая автопилотируемые автомобили) составит 29% (3,3 млрд) от общего числа мобильных подключений. В 2016 г. этот показатель составлял 5%, т.е. подключение M2M — самый быстрорастущий тип мобильных соединений.

С Android в качестве базовой ОС, Volvo рассчитывает «повысить скорость и гибкость разработки и позволить клиентам персонализировать свое общение с подключенным автомобилем». Android появится в новых моделях Volvo в течение двух лет. В результате клиенты компании получат доступ к широкому спектру ПО и сервисов, включая приложения для Android от Google, Volvo и третьих сторон. При этом, используя «богатую экосистему Android», Volvo планирует сохранить свой интерфейс пользователя, пояснил Хенрик Грин [Henrik



» В последних моделях Volvo приложение для определения местоположения Google Local Search сможет работать с новой версией навигационной системы Sensus Navigation.

Green], старший вице-президент по исследованиям и разработкам Volvo Car Group. Представители Volvo также заявили об отдельном сотрудничестве с Google в целях адаптации Google Local Search к последним моделям Volvo. LXF

Сравнение

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

Лучший дистрибутив KDE

Лихо скакавший с одного дистрибутива на другой Маянк Шарма оробел, когда пришлось отойти от дистрибутивов Gnome... но скоро освоился.



Про наш тест...

Ряд главных настольных дистрибутивов предлагают на выбор несколько рабочих столов. И можно установить KDE поверх практически любого настольного дистрибутива. Однако в нашем Сравнении мы сосредоточились на дистрибутивах, загружаемых прямо в рабочий стол KDE. Чтобы проект прошел, дистрибутив должен располагать официальной разновидностью KDE.

Сам KDE выпускает среду KDE Neon Live, но мы не включили ее, поскольку KDE Neon, по его собственному признанию, не считает себя дистрибутивом — скорее, это выставочная витрина для рабочего стола KDE.

Мы сравним дистрибутивы по таким параметрам, как пакеты по умолчанию и механизмы обновления. Основными критериями при выборе победителя, однако, будут внешний вид и впечатление от работы с дистрибутивом. Тот, который поможет пользователям узнать лучшее в KDE, получит по очкам предпочтение перед другими.

Наша подборка

- » Chakra
- » KaOS
- » Manjaro
- » Maui
- » Netrunner

Дистрибутивы Linux поставляются с определенным рабочим столом, который, правда, легко сменить на другой; но большинство пользователей предпочитают придерживаться выданного по умолчанию. Можно утверждать, что Gnome — господствующий рабочий стол по умолчанию и так или иначе присутствует в большом числе дистрибутивов. Вы найдете среды на базе Gnome в широком спектре проектов, от обычных настольных до специализированных сборок.

И хотя нельзя сказать, что у KDE недостаточно пользователей, этот рабочий стол в общем не получает того внимания, которого он явно заслуживает. Вероятность

того, что новичок в Linux начнет с KDE, довольно-таки мала, с учетом того факта, что на большинстве путей, ведущих к Linux, используется по умолчанию среда рабочего стола не-KDE. Кроме того, исторически так сложилось, что рабочий стол KDE более авантюрный, и он умудрялся бесить опытных пользователей Linux задолго до того, как Gnome и Unity начали их переманивать. И хотя рабочий стол значительно вырос

по сравнению с днями KDE 4, часть пользователей ушла в другие уголки и не в курсе насчет новых интересных разработок.

Стремясь исправить эту оплошность, мы рассмотрим некоторые дистрибутивы, сосредоточившиеся на рабочем столе KDE. В нашем Сравнении вы найдете дистрибутивы, пригодные для начинающих пользователей, и такие, которые впечатлят ветеранов своими зрелыми платформами.

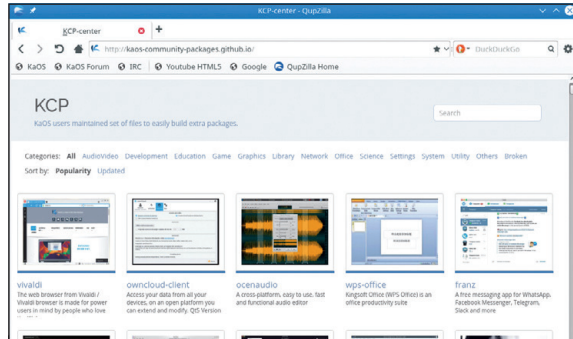
Рабочий стол значительно вырос по сравнению с днями KDE 4 насчет новых разработок.

Приложения по умолчанию

Разбавляют ли они подборку приложениями не-KDE?

Подборка приложений проекта KDE настолько обширна, что перед самым выходом KDE 4 официальные приложения были переведены в отдельный продукт KDE, KDE Applications. Помимо обычных рабочих приложений типа текстовых редакторов и просмотрщиков изображений, там также есть альтернативы web-браузерам и офисным пакетам. В стремлении дать вам шансы познакомиться с лучшим в KDE одни дистрибутивы включают только приложения KDE, а другие вместо них включают их более популярные альтернативы.

Chakra явно позиционирует себя как дистрибутив KDE, и по этой причине по умолчанию предлагает приложения KDE. Наиболее интересные такие приложения — *Krita*, *Karbon*, *Kget*, *Kdenlive*, *digiKam*, *Kmail*, пакет *Calligra Office*, web-браузер *Qupzilla* и утилита скриншотов *Spectacle*. Другие приложения на *Qt* — медиа-плеер *Vomli* и музыкальный плеер *Tomahawk*. Имеется также сервис менеджера хранилища, чтобы открывать файлы в *Dropbox*, *YouSendIt*, *Google Drive*, *Vox* и любом месте *WebDAV*. *KaOS* тоже предлагает приложения KDE по умолчанию. Помимо обычных, есть еще *Quassel IRC*, клиент для *MPD Cantata*, web-камера



» Вы можете одним нажатием установить самые разные поддерживаемые сообществом пакеты с сайта *KaOS*.

Komoso, медиа-плеер *mpv*, *SimpleScreenRecorder*, *SMPlayer*, *SMTube* и *Qupzilla*. В отличие от *Chakra*, *KaOS* включает менеджер загрузки *FatRat*, клиент *Seafile* и проприетарный клиент *Skype*.

Manjaro предлагает официальный релиз KDE, хотя его флажманское предложение — релиз *Xfce*. Поэтому *Manjaro* снисходительнее к не-KDE приложениям, чем два предыдущих. Т.е. здесь солидная подборка приложений KDE, но большая часть рабочих приложений — популярные программы, а не их KDE-альтернативы. Вы найдете в *Manjaro LibreOffice*, *VLC*, *Firefox*, *Thunderbird* и клиент *Steam*. Аналогично, *Maui* и *Netrunner* комбинируют приложения KDE

с популярными открытыми приложениями, такими как *LibreOffice*, *Firefox*, *Thunderbird*, *GIMP* и *VLC*. Также только эти оба включают *Ndiswrapper* для установки беспроводных драйверов *Windows*. Помимо обычных приложений KDE, в них также есть несколько проприетарных приложений, в том числе клиент *Steam*, *VirtualBox* и *Skype*.

Чтобы освоиться с букетом приложений KDE, надо к ним привыкнуть, но так как цель нашего Сравнения — выбор дистрибутива, который лучше всего освещает все аспекты KDE, мы присудим больше баллов тем проектам, которые придерживаются приложений KDE, не заменяя их популярными и знакомыми альтернативами.

Вердикт

- Chakra ★★★★★
- KaOS ★★★★★
- Maui ★★★★★
- Netrunner ★★★★★
- Manjaro ★★★★★

» *Chakra* и *KaOS* предлагают чистый рабочий стол KDE.

Политика пакетирования

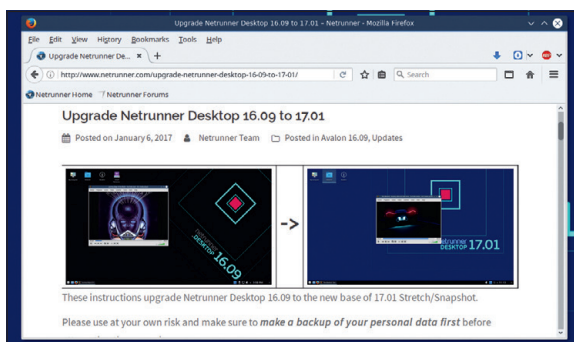
Как они управляют своими репозиториями?

Все пакеты *Chakra* распределяются по нескольким репозиториям, например, «настольный [desktop]», с пакетами KDE, и «gtk», с различными приложениями *GTK*. Далее имеется репозиторий «нестабильные [unstable]», с самыми современными пакетами. Есть также репозиторий, поддерживаемый сообществом, вдохновленный *Arch User Repository*.

KaOS сначала помещает пакеты в репозиторий сборки, а после тестирования переводит их в один из трех пользовательских репозиториях — *core* [ядро], *main* [основное] и *apps* [приложения]. *Core* — возобновляемый, но только после того, как все компоненты будут тщательно протестированы. Хотя *apps* полностью возобновляемый, многие пакеты из репозитория *main*,

содержащие библиотеки, драйверы и прошивки, являются возобновляемыми и доступными для пользователя после десяти дней тестирования. Как и в *Chakra*, в *KaOS* есть и поддерживаемый сообществом репозиторий. Пакеты *Arch* в конечном итоге оказываются в «нестабильном» репозитории *Manjaro*, и затем тестируются в течение недели, после чего переводятся в стабильный репозиторий.

Maui базируется на *Ubuntu Xenial* и использует его репозитории для основных компонентов, которые заодно размещают пакеты KDE *Neon*. Для самых новых компонентов у *Maui* имеются два репозитория обратных портов. В *Netrunner* применяется такой же подход, что и в *Maui*. Он основывается на снимке для локального развертывания ветви *Debian Testing*. Вы можете активировать для себя репозитории *Backports* и *Debian Testing*, чтобы получать постоянно тестируемые обновления.



» Хотя *Netrunner* не является возобновляемым релизом, разработчики выложили скрипт *Bash*, который помогает обновиться до новой версии.

Вердикт

- Chakra ★★★★★
- KaOS ★★★★★
- Manjaro ★★★★★
- Maui ★★★★★
- Netrunner ★★★★★

» Все дистрибутивы тщательно добывают вышедшие программы из их репозиториях.

Опыт KDE

Насколько они KDE-шные?

Поскольку здесь все кандидаты на тестирование используют рабочий стол KDE, на первый взгляд они кажутся одинаковыми. Вдобавок у них совпадают списки приложений, включенных в пакеты по умолчанию, что увеличивает сходство.

На деле их отличают друг от друга их цель и взаимодействие с пользователем, к которому они стремятся. Всё это становится очевидным по мере работы с дистрибутивом.

Одни дистрибутивы нацелены на конкретный тип пользователей, тогда как другие являются

достаточно гибкими, чтобы устраивать большее количество пользователей. В данном разделе мы оценим, какой дистрибутив лучше всего использует имеющиеся в его распоряжении инструменты, чтобы предложить более отлаженный, гладкий и удобный рабочий стол KDE.

Chakra Linux ★★★★★

Дистрибутив позиционируется как сугубо KDE, и вам придется потрудиться, чтобы найти в его меню не-KDE приложение. Разработчики также убрали зависимости *GTK+* из популярных пакетов на базе *Gnome*, устанавливаемых из его репозитория. *Chakra* выглядит располагающе и использует менеджер отображения *SDDM*, весьма симпатичный и хорошо интегрирующийся с рабочим столом *Plasma*. Помимо обычных приложений KDE, дистрибутив включает несколько таких, которые вы больше нигде не найдете: например, медиа-плеер *Bomi*, музыкальный плеер *Tomahawk* и менеджер сервисов хранения *KDE Storage*. Все, что относится к проекту *Chakra*, например, ссылки на документацию, форумы, баг-трекеры, журнал изменений, репозиторий кода и т.д., тщательно организованы внутри подменю *Chakra* в меню *Application*. Целью, очевидно, является обеспечение пользователей комплексной работой с KDE.



KaOS ★★★★★

Разработчики *KaOS* утверждают, что хотя разнообразие программ с открытым кодом — это хорошо, дистрибутивы всё же должны сами делать ряд выборов, которые считают наилучшими для своих пользователей. Исходя из этого, *KaOS* на данный момент основан на *Linux*, но постоянно оценивает и ядро *illumos*. А вот чего дистрибутив придерживается твердо, так это рабочий стол *KDE Plasma* и инструментарий *Qt*. На случай, когда эквивалент *KDE/Qt* не предлагает определенных функций, дистрибутив держит в своих репозиториях пакеты *GTK*. *KaOS* также единственный дистрибутив, который включает медиа-плеер *mpv* и приложения *Kamoso*, *SMPlayer* и *SMTube*. Особенно радует включение программы приветствия рабочего стола *Kaplan*, позволяющей настроить разные аспекты установки, хотя размещение меню приложений на правой стороне экрана требует привыкания, а способа его переноса в другое место нет.

Простота установки

Насколько легко их установить?

Это еще один из тех параметров, который на самом деле не является решающим фактором при выборе одного из этих кандидатов. Причина в том, что все дистрибутивы в нашем тесте используют независимую от дистрибутива программу установки *Calamares*. По сути, *Calamares* — это среда, которую дистрибутивы могут индивидуально настроить для своих целевых пользователей. Программа установки довольно интуитивна и визуально привлекательна, и работает без особой помпы. Ее функция разбиения диска

на разделы поддерживает и ручное, и автоматизированное разбиение. Она также включает очень удобную опцию *Replace Partition*, позволяющую повторно использовать раздел для смены дистрибутива. Короче, он лучше всего подходит для простых установок и пока не поддерживает расширенных структур типа разделов *RAID* и *LVM*.

Различие в реализации программы установки *Calamares* в разных дистрибутивах чисто косметическое. Еще одно отличие состоит в том, что некоторые дистрибутивы выдают вам опцию шифрования раздела

root, а некоторые — нет. Реализация программы установки в *Chakra*, *Maui* и *Netrunner* подпадает под последнюю категорию, и не предлагает опции шифрования вновь созданного раздела. И наоборот, *Manjaro* и *KaOS* позволяют зашифровать раздел, на который вы планируете установить соответствующий дистрибутив. Разработчики *KaOS* также настроили *Calamares* для лучшего определения других установленных операционных систем и дистрибутивов и их идентификации на стадии разбиения на разделы.

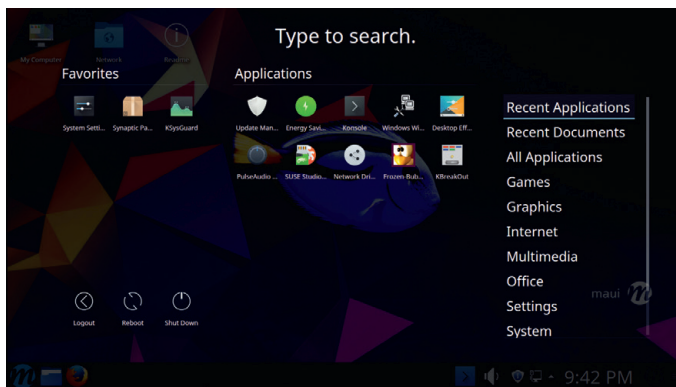
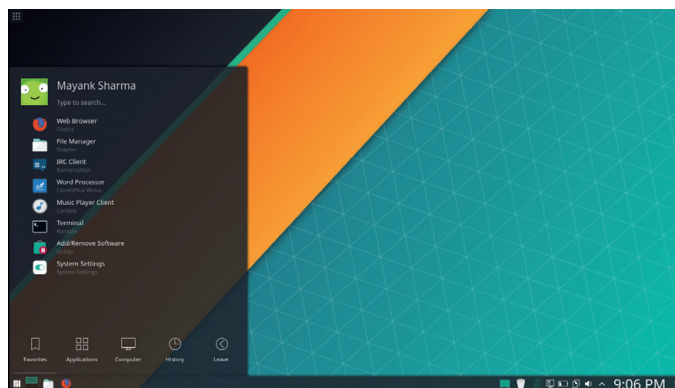
Вердикт

- KaOS** ★★★★★
- Manjaro** ★★★★★
- Chakra** ★★★★★
- Maui** ★★★★★
- Netrunner** ★★★★★

» Все дистрибутивы равны, поскольку все используют установщик *Calamares*.

Manjaro ★★★★★

Редакция KDE не флагманская для Manjaro, и поэтому дистрибутив более открыт для использования не-KDE приложений, и тем не менее, за исключением популярных рабочих приложений типа *LibreOffice* и *Firefox*, изобилует приложениями KDE. Manjaro — полностью возобновляемый дистрибутив, и разработчики постарались, чтобы их пользователи среди первых могли познакомиться с самым последним релизом компонентов KDE, включая рабочий стол Plasma, Apps и Framework. Manjaro — единственный дистрибутив, предлагающий инструмент *Kleopatra* для управления сертификатами. В дистрибутиве есть свой набор индивидуальных инструментов, размещенных в менеджере настроек Manjaro, который помогает пользователям выполнять административные задачи, такие как переключение между имеющимися ядрами. К тому же Manjaro привлекателен внешне, и его интерфейс достаточно красив.

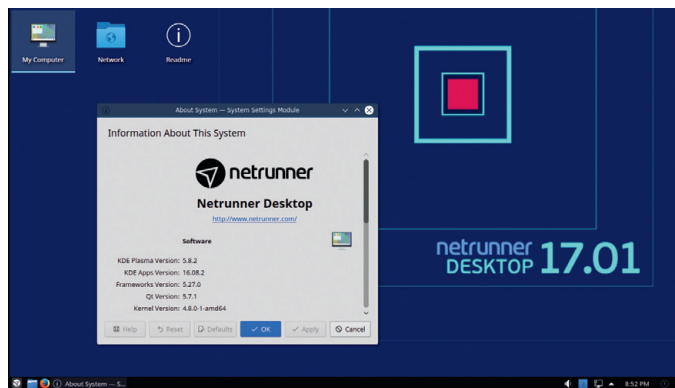


Maui ★★★★★

Maui загружает сильно индивидуализированный рабочий стол, который, вероятно, разработан для больших дисплеев. На ноутбуках без разрешения FullHD значки кажутся огромными и не позволяют рабочему столу выглядеть столь же гладко, как в некоторых других вариантах KDE. Например, некоторые особо длинные названия приложений обрезаны, что выглядит весьма непрофессионально и дезориентирующе. По умолчанию Maui использует полноэкранный программу запуска приложений KDE, очень похожую на Gnome 3 Activities и Ubuntu Unity. Однако, как и другие дистрибутивы, Maui также предлагает три других программы запуска. Он использует *Update Manager* из Mint, что очень упрощает обновление установки Maui. Дистрибутив содержит интересный микс приложений KDE, не-KDE и проприетарных приложений, и разработчики постарались интегрировать в Plasma «неродные» приложения.

Netrunner ★★★★★

По умолчанию Netrunner использует традиционную программу запуска приложений с каскадными меню, но, как и в других, можно выбрать альтернативу. Благодаря поднастройке и темам, он выглядит гармонично, несмотря на наличие ряда проприетарных и не-KDE приложений, как и в Maui. Настройка распространяется и на ключевые приложения KDE, такие, как *Dolphin* и окно *System Settings* [Системные настройки]. Например, помимо обычных модулей настройки, окно *System Settings* также включает раздел System administration [Администрирование системы]. Есть также раздел Advanced [Расширенный], откуда можно управлять такими сервисами Plasma, как *KRunner* и *Akonadi*. Как и в Maui, обязанность обновления выполняется *Update Manager* из Mint, но разработчикам стоит передумать свое решение использовать *Synaptic* для управления пакетами вместо одного из более интуитивных интерфейсов.



Документация и поддержка

Куда обратиться за помощью?

Первичный режим по предоставлению помощи и поддержки для всех тестируемых нами дистрибутивов — скромные форумы. Некоторые из них, например, у Chakra, довольно активны, а Manjaro вообще уникален, потому что располагает форумами на нескольких языках.

Maui поддерживает блог для анонсирования новых релизов. На рабочем столе в Maui и Netrunner имеется также значок README. Тот, что в Maui, отображает краткую информацию о текущем релизе, а тот,

что в Netrunner, отображает информацию, крайне полезную после установки; и дистрибутив также предлагает иллюстрированный справочник по установке.

Chakra включает иллюстрированное руководство для начинающих и wiki со статьями, которые помогут в установке и управлении дистрибутивом. Однако учтите, что хотя многие статьи довольно подробны, некоторые, как, например, статья об *Octopi*, состоят всего из одной строки.

На сайте KaOS вы обнаружите разнообразные краткие руководства пользователя,

например, по *raspm*, по переключению на ядро Linux-next, установке KaOS на SSD и т.п. Manjaro опережает всех благодаря каналу на Vimeo, где есть более двух десятков видео по разным аспектам дистрибутива и его разработки. Его wiki достаточно подробна и хорошо категоризирована, и поможет пользователям установить, управлять и решать проблемы с их установкой. Помимо форумов, у Manjaro также имеются многоязычные каналы IRC и несколько списков рассылки для обсуждения разных аспектов проекта.

Вердикт

- Manjaro ★★★★★
- Chakra ★★★★★
- KaOS ★★★★★
- Netrunner ★★★★★
- Maui ★★★★★

» У всех дистрибутивов есть активное сообщество, которое всегда поможет.

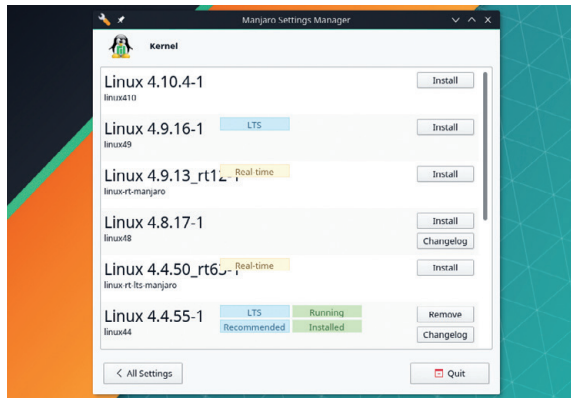
Степень настраиваемости

Что делает их особенными?

Ни один из дистрибутивов в этом Сравнении не предлагает стандартного рабочего стола KDE. Однако степень настраиваемости у них разная. Некоторые даже настроили раскладку основных приложений KDE, например, менеджера файлов *Dolphin*, чтобы сделать их привлекательнее для предполагаемой базы пользователей.

Все дистрибутивы предлагают разные типы меню. Есть простое меню приложений, всплывающие меню каскадом, используемые по умолчанию несколькими дистрибутивами, а также полноэкранный Application Dashboard в стиле Unity/Gnome3.

Maui, Netrunner и KaOS не предусматривают индивидуальных инструментов для содействия администрированию, которое отнесено к окну KDE System Settings. Maui и Netrunner настроили *Firefox* на включение некоторых дополнений, таких, как AdBlock Plus и программа скачивания видео Ant. С другой стороны, KaOS — единственный дистрибутив, размещающий панель с правой стороны экрана вместо обычного размещения внизу. Он также предлагает опцию запуска сессии Plasma Wayland на экране



» В Manjaro Settings Manager есть демон для уведомления пользователей о наличии нового ядра для установки.

приглашения, и использует мастер первого запуска *Kapitan* для настройки разных аспектов вновь установленной системы, таких как поведение мыши и т.п.

Manjaro включает *Manjaro Settings Manager (MSM)*, помогающий пользователям переключаться между имеющимися ядрами. Инструмент также автоматически устанавливает все необходимые модули ядра для выбранного ядра. *MSM* включает распознавание модулей оборудования, позволяя выбирать между проприетарными

и свободными драйверами для подключения оборудования, в том числе для видеокарт. Ядро Chakra настроено с помощью некоторых опций укрепления MAC, в том числе *Tomoyo* и *AppArmor*. Дистрибутив предлагает также графический скрипт *minibackup* для резервного копирования таких персональных данных, как адресная книга, SSH-ключи электронной почты и настройки приложений. Есть еще модуль редактора репозитория Chakra, который облегчит вам управление репозиториями.

Вердикт

- Chakra ★★★★★
- Manjaro ★★★★★
- KaOS ★★★★★
- Maui ★★★★★
- Netrunner ★★★★★

» Manjaro и Chakra идут на шаг вперед благодаря своим индивидуальным инструментам.

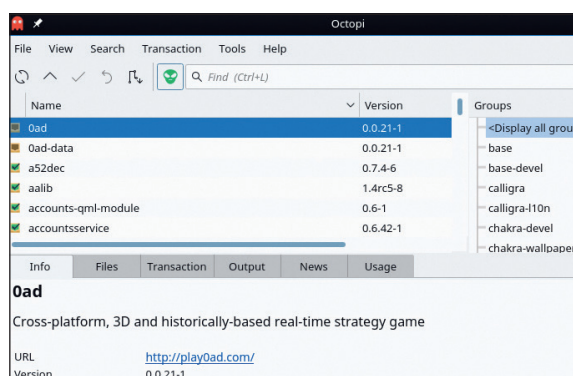
Политика и средства обновления

Идти в ногу со временем.

Большинство наших дистрибутивов основано на Arch Linux. Двумя исключениями являются Maui, на базе Ubuntu, и Netrunner, на базе образа Debian Testing. Все проекты используют для получения обновлений инструменты того дистрибутива, на который опираются. Дистрибутивы на базе Arch используют для установки отдельных пакетов и обновления установки *Ostropi*, графический интерфейс к *raspm*, менеджеру пакетов Arch. Maui и Netrunner используют для установки отдельных пакетов *Synaptic* и *Update Manager* от Mint, а не от Ubuntu.

Явным различием между двумя системами является то, что процесс обновления дистрибутивов, применяемый в *raspm*, более многословен, и иногда требует подтверждения вручную. Процессы обновления списка зеркал в разных дистрибутивах тоже разные.

Maui по умолчанию следует стандартному циклу релизов. Если вам нужны суперсовременные программы, вы можете



» Зеленая голова инопланетянина в *Ostropi* означает, что дистрибутив включает *Yaourt*, позволяющий установить пакеты из Arch User Repository.

включить свою установку в частично возобновляемый релиз, активировав канал обратных портов Xenial. Также вы можете запустить самый последний релиз Netrunner как стабильную установку или включить репозиторий тестирования Debian.

Manjaro, напротив, полностью возобновляемый релиз, как и сам Arch. KaOS — тоже. Разработчики сделали так, что даже если пакет не обновлялся в течение года,

он будет перекомпилирован, чтобы хорошо интегрироваться с остальной системой. По части ядра дистрибутив предлагает две опции. Есть стабильное ядро Linux и полностью возобновляемое ядро Linux-next. Разработчики Chakra выбрали полувозобновляемую модель, предоставляя стабильные основные компоненты вместе со свежими приложениями и обновлениями безопасности, т.е. лучшее из обеих областей.

Вердикт

- Chakra ★★★★★
- KaOS ★★★★★
- Manjaro ★★★★★
- Maui ★★★★★
- Netrunner ★★★★★

» KaOS и Chakra ищут баланс суперсовременных программ на стабильном ядре.

Лучший дистрибутив KDE

Вердикт

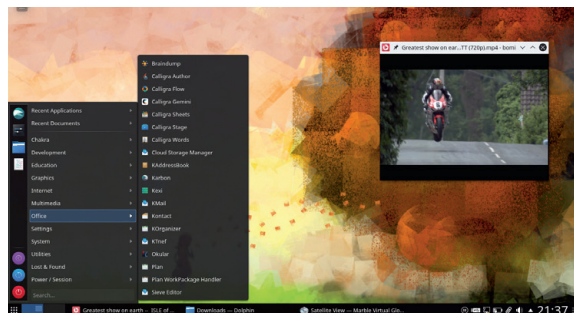
Огромное сходство между нашими кандидатами осложнило данное Сравнение. Одним из основных критериев является количество не-KDE приложений, включенных по умолчанию. Поскольку мы ищем лучший дистрибутив KDE, мы благосклонно смотрели на те из них, которые предпочитают приложения KDE другим популярным приложениям.

Maui замыкает ряд, поскольку на экранах без FullHD выглядит хуже своих товарищей. Он набрал очки за интересный микс из приложений KDE и проприетарных приложений, но проигрывает по причине отсутствия документации и из-за своего внешнего вида. Его старший родич, Netrunner — хорошо собранный дистрибутив, который отлично справляется с задачей демонстрации KDE, но тоже проигрывает из-за своего решения использовать популярные приложения вместо их альтернатив KDE.

Фактически, единственным реальным различием между Maui и Netrunner является дистрибутив, на котором они основаны.

Maui — продолжение оригинального дистрибутива Netrunner на базе Kubuntu, но теперь основан на KDE Neon и получает свои стабильные основные компоненты из релизов Ubuntu LTS. Netrunner теперь основан на ветви Debian Testing, которая закончится следующим стабильным Debian под названием Stretch. Manjaro поднялся на пьедестал, потому что это один из немногих проектов, разработанных не-корпоративной командой, который по-прежнему доступен для 32- и 64-битных компьютеров. KDE не является его первым рабочим столом по выбору, и дистрибутив по умолчанию включает несколько не-KDE приложений.

Два лидирующих дистрибутива доступны только для 64-битной архитектуры, и идут голова к голове. KaOS набирает очки за включение большего числа сторонних приложений по умолчанию, чем Chakra, таких как Skype, и за опцию настройки. Он к тому же предлагает опцию



запуска Plasma поверх Wayland. KaOS отлично демонстрирует философию KDE благодаря приложению приветствия, позволяющему пользователям выбрать тип меню, тему и т.п. Но проигрывает из-за размещения панели приложений на правой стороне экрана, что требует привыкания, и нет явного способа это изменить. В итоге остается Chakra — он отлично справляется с задачей представления KDE и имеет достойную подборку приложений KDE. В отличие от соперников, Chakra умудряется поддерживать должный баланс между стабильным ядром и суперсовременными приложениями.

» Chakra также является одним из самых удобных в использовании дистрибутивов по умолчанию.

Отличное представление KDE и впечатляющая подборка приложений KDE.

I Chakra ★★★★★
 Версия: 2017.03 Сайт: <https://chakralinux.org> Лицензия: GPL и др.
 » Чистый дистрибутив KDE. Суперсовременные приложения на стабильной платформе.

IV Netrunner ★★★★★★
 Версия: 17.01.2 Сайт: www.netrunner.com Лицензия: GPL и др.
 » Включает разумное число KDE, не-KDE и проприетарных приложений.

II KaOS ★★★★★★
 Версия: 2017.02 Сайт: <https://kaosx.us> Лицензия: GPL и др.
 » Еще один тщательно созданный дистрибутив KDE, применять его — сплошное удовольствие.

V Maui ★★★★★★
 Версия: 17.03 Сайт: <https://mauilinux.org> Лицензия: GPL и др.
 » Основанная на KDE Neon версия Netrunner с теми же приложениями.

III Manjaro ★★★★★★
 Версия: 17.0 Сайт: <https://manjaro.org> Лицензия: GPL и др.
 » Полный релиз с использованием популярных приложений вместо их альтернатив KDE.

Обратная связь
 Мы неправы? Правы? А какой ваш любимый дистрибутив KDE? Поделитесь своим мнением на lxformat@futurenet.com.

Рассмотрите также...

В популярных и более мелких дистрибутивах с рабочим столом KDE нехватки нет. К популярным относятся Kubuntu, openSUSE и Mageia, и у всех у них рабочий стол KDE сильно модифицирован. Вы можете попробовать KDE на Debian и Linux Mint; разновидность KDE в Fedora использует практически неизменный рабочий

стол KDE, и то же самое относится к Korora KDE, но с большим количеством приложений по умолчанию. Два менее крупных проекта — SparkyLinux на базе Debian и Antergos на базе Arch.

Вы также можете использовать KDE прямо от разработчиков в KDE Neon. Хотя этот дистрибутив на базе Ubuntu включает программу установки,

разработчики отвели ему роль всего лишь демонстрационной платформы, а не дистрибутива для повседневного использования.

Если вы не категорический противник операционных систем не-Linux, несколько BSD также включают рабочий стол KDE, в том числе TrueOS, FreeBSD и OpenBSD. **LXF**

Ubuntu 17.04

Прошло уже полгода, как Yakkety Yak вышел за ворота системы сборки, а значит, пора выпускать на свободу Zesty Zapus.

Весна окончательно пришла. И это может означать только одно. Вернее, это может означать одно из множества вещей — сенную лихорадку, юношескую влюбленность, цветение, празднование Вальпургиевой ночи на Лысой горе; но если оставить в стороне все эти мелочи, то на самом деле весна означает новый релиз Ubuntu под номером и названием 17.04 и Zesty Zapus соответственно. Мы так хотели включить его на наш диск, что до последнего отодвигали срок сдачи материалов в набор, заискивали перед польскими репликаторами диска и заставили этого автора засиживаться допоздна, хлебая кофе литрами.

Хотя не-LTS релизы не рекомендованы для рабочих систем, тестируются они тщательно и отлично подходят для показа, в каком направлении будет двигаться LTS следующего года. Пользователи предыдущих релизов найдут обычное обновление пакетов и полностью обновленное ядро. Однако в этом релизе ощущается также усиливающееся течение в сторону новых технологий,

На самом деле весна означает новый релиз Ubuntu.

Хотя не-LTS релизы не рекомендованы для рабочих систем, тестируются они тщательно и отлично подходят для показа, в каком направлении будет двигаться LTS следующего года. Пользователи предыдущих релизов найдут обычное обновление пакетов и полностью обновленное ядро. Однако в этом релизе ощущается также усиливающееся течение в сторону новых технологий,

а именно Snaps (атомарный формат пакетов) и Unity 8. Они не являются обязательными (и в случае последнего даже пока что не готовы, как мы увидим), но те, кто решит использовать их на ранней стадии, получат большое удовольствие от созерцания потенциала этих новых инструментов. Хотя времена сделок с рекламодателями и сотрясающих основы изменений UI, к счастью, остались позади [Ред.: —Ха!], в 17.04 есть что исследовать; этим мы и займемся в нашей статье. Вы увидите обзор того, что изменилось, а что осталось неизменным; советы по улучшению установки по умолчанию; и мы заглянем под капот, чтобы показать вам ряд продвинутых настроек. Наш храбрый читатель, вперед!

Что же несет нам Zapus?

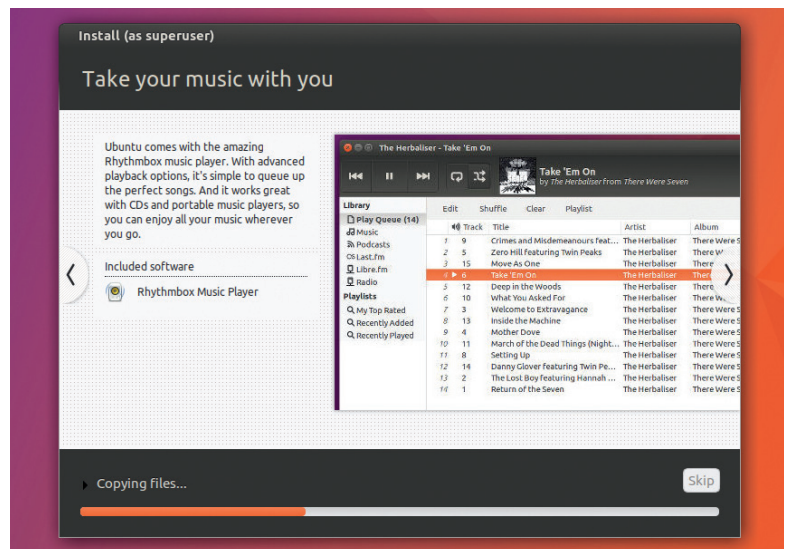
Изменения в 17.04 лучше всего описываются как «Эволюция, не революция», однако в этом релизе всё равно немало интересного.

Наверняка любители Ubuntu больше всего хотят увидеть в этом релизе, что происходит с Unity 8, и следующий разворот мы отвели работающей на *Mir*, независимой от аппаратной части (и недолго пока существующей в этом мире) среде рабочего стола. Но помимо этого, в Ubuntu 17.04 немало нового и интересного.

Однако большая часть всего этого находится не на поверхности — сессия Unity 7 по умолчанию выглядит практически так же, как в прошлом году; пользователям не придется заново учиться управлять пакетами (хотя эта опция доступна), а выбор установленных по умолчанию приложений остался неизменным с 16.10. Однако начнем с самого основания: у нас совершенно новое ядро, 4.10, новая версия *systemd*, которая теперь выполняет обязанности по разрешению имен через *systemd-resolved*, новая версия сервера *X.org* и некоторые вручную отобранные компоненты из *Gnome 3.24*.

Более новое ядро означает лучшую поддержку оборудования; в частности, с этим ядром новые процессоры Ryzen от AMD будут работать намного эффективнее. Кстати об AMD: есть поддержка их архитектуры Polaris 12, и информация о вентиляторе теперь выводится через *hwmon*. Есть также экспериментальная поддержка разблокирования частот Boost на Nvidia GPU с помощью свободного драйвера Nouveau. Еще важнее то, что пользователи новых карт Nvidia теперь могут управлять встроенными LED через интерфейс *sysfs*. Плюс к этому, предусмотрена поддержка менее дорогого оборудования — да, есть шанс, что только что приобретенный вами новый USB-брелок тоже поддерживается!

Одно изменение, нарушающее общую тенденцию, заключается в отмене разделов подкачки. Они были основным компонентом установок Linux с самого начала — когда регулярно приходилось сталкиваться с нехваткой памяти. Вместо этого установка по умолчанию (если только у вас уже не создан раздел подкачки: в этом случае вы можете предпочесть продолжить его использование) применяет файлы подкачки по запросу, что несколько эффективнее. Часто повторяемый, но редко используемый совет сделать раздел подкачки вдвое больше объема ОЗУ становится совершенно бессмысленным в эпоху, когда 8 ГБ ОЗУ стали нормой: на 16 ГБ вполне мог бы улезть целый дистрибутив.



➤ Это cameo пионера джаз-хип-хопа The Herbaliser определенно привлекло наше внимание. Кстати, чистая установка занимает чуть более 4 Гб.

Панацея для принтеров

Печать в Linux всегда была задачей намного сложнее, чем следовало бы. Печальная истина в том, что весьма часто люди печатают в PDF и переносят этот PDF на компьютер с Windows или используют Google Cloud Print, когда хотят получить бумажный экземпляр. Как и в случае с другим оборудованием, ситуация еще более запутывается из-за того, что производители предлагают на своих сайтах драйверы, которые ни в жизнь не соглашаются работать с современным дистрибутивом Linux и даже имеют неплохие шансы окончательно его доломать.

Конечно, драйверы доступны в пакете CUPS, но с 17.04 начинают печатать без драйверов и, возможно, улучшение текущей ситуации. Это работает только с некоторыми моделями, в том числе с теми, которые поддерживают IPP Everywhere (см. <http://www.pwg.org/dynamo/eveprinters.php>) и Apple Airprint, но ведь это только начало. ➤

Улыбнитесь, снимают

Снимки (или snap-пакеты, или назовите их сами по своему усмотрению) и инструменты для работы с ними доступны в Ubuntu с 16.04.

Они решают ряд проблем с текущими методами пакетирования. Прежде всего — благодаря новомодным контейнерам, они устанавливаются изолированно от остальной системы. И даже если вы установите злокозненный снимок, большого вреда он причинить не сможет. Кроме того, снимки, по крайней мере, в одном смысле, куда более переносимы, чем традиционные пакеты *.deb*.

Их можно установить на любой дистрибутив с поддержкой snaps, поскольку они содержат все свои

зависимости. Конечно, из-за этого они объемнее традиционных пакетов, но зато намного упрощается процесс пакетирования: разработчикам достаточно сделать пакеты один раз для всех дистрибутивов. Снимки легко обновлять, поэтому и обычные проблемы устаревших пакетов в репозиториях становятся неактуальными. Например, если вы хотите иметь всегда обновленную версию безопасного приложения обмена сообщениями *Telegram*, то всё, что вам нужно —

```
$ sudo snap install telegram-latest
```

На сайте UAppExplorer <https://uappexplorer.com/apps?type=snap> есть масса всяких снимков.

Один из наших фаворитов — мощный OhMyGiraffe, который мы оставим вам для самостоятельного исследования. Снимки — не единственный способ пакетирования, не имеющий зависимостей и независимый от оборудования: уже некоторое время существуют также форматы Flatpak (бывший *xdgapp*) и *AppImage*.

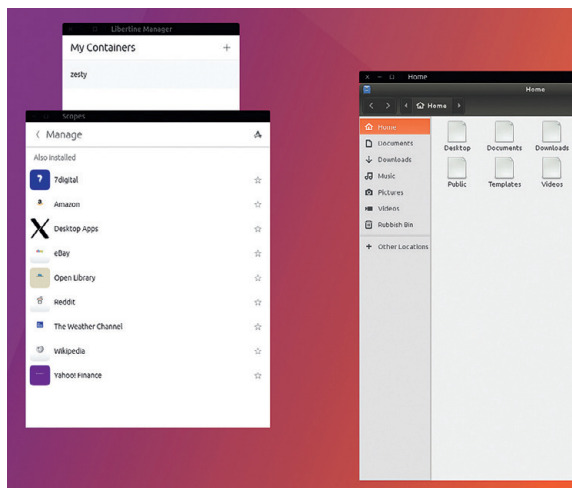
Flatpaks — подход к созданию пакетов, более ориентированный на настольные системы — также поддерживается приложениями. Надо надеяться, что эти форматы смогут гармонично сосуществовать и намного упростят жизнь разработчиков, не вызывая фрагментации.

Рабочий стол Unity 8

Как лучше описывается Unity 8: наконец-то избавленный от мучений или безвременно ушедший из этого мира? Решайте сами.

Утонченные любители музыки помнят судьбоносный альбом Queen *Latifah U. N.I. T.Y.* от 1993 г., который бросил вызов господствующим в рэпе того времени женоненавистническим тенденциям. Среда рабочего стола Unity с момента своего появления в 2010 г. тоже постоянно бросает вызов: терпению пользователей, их представлениям о конфиденциальности и о том, что боковая панель запуска считалась плохой идеей с того времени, как в 2000 г. ее испробовал *Microsoft Office*. К настоящему моменту большая часть криков и шума семилетней давности уже затихла. Unity, как минимум, не в последнюю очередь благодаря постоянной панели приложений, сохраняет хотя бы намек на идеал традиционного рабочего стола, который Gnome 3 благополучно вышвырнул в окно.

В целом люди привыкли к той мысли, что навигация по каскаду меню в поисках нужной программы — не самый эффективный способ работы: несколько прицельных ударов по клавишам [Ред.: — А можно и то, и другое вместе?] способны перенести вас в желаемое место куда быстрее. Пользователи *Emacs* и управляемых с клавиатуры менеджеров окон твердили нам об этом не один год, но Unity не заставляет вас учить гимнастические упражнения для трех или четырех пальцев ради запуска приложения и дает вам нечто, выглядящее современно и удобно. Усадите новичка за базовую сессию i3, и вы поймете, что мы имеем в виду. Unity пошла на уступки многим из наиболее частых нареканий (исключая кнопки окна с левой стороны) за несколько лет, и в результате вышел

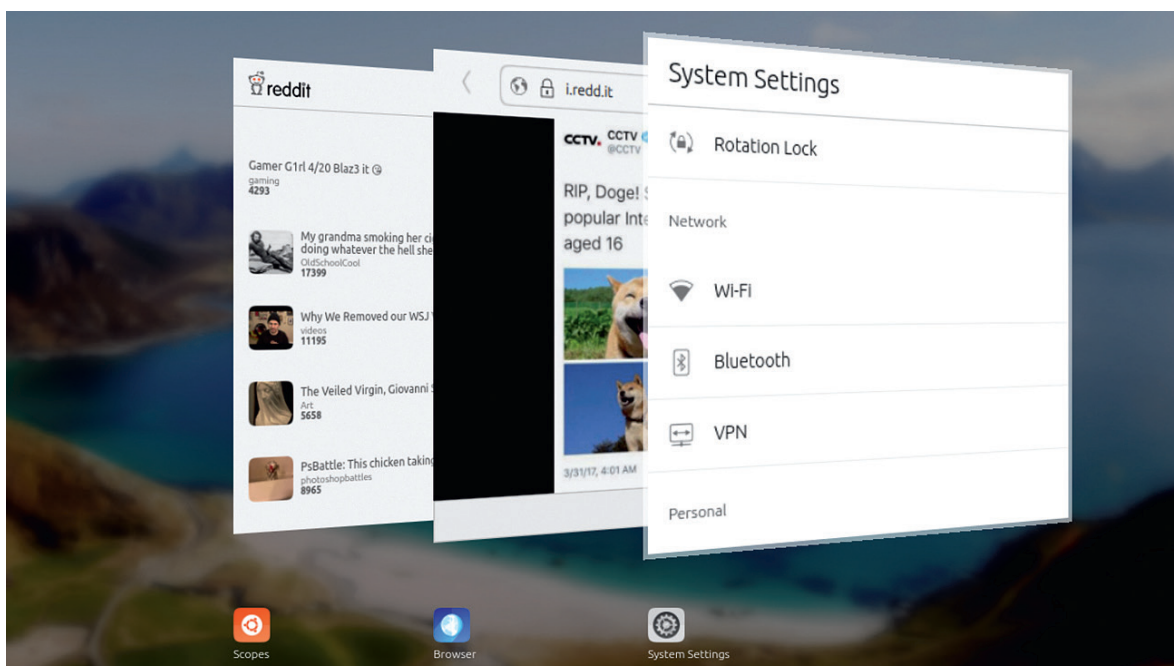


► Приложения в старом стиле X11, там, где они работают, выглядят странно с двойными строками заголовка. Кажется также, что Files растерял свои значки.

рабочий стол, который многие как минимум могут терпеть, а некоторые даже используют с удовольствием.

Если не брать во внимание некоторую неуверенность в том, где именно должны быть панели меню, Unity 7 не очень изменилась за последние три года, и рабочий стол по умолчанию 17.04 с этим не спорит. Unity 7 благополучно законсервирована, и будущее за Unity 8. По крайней мере, так было вплоть до самого момента готовности данного материала, когда Марк Шаттлворт [Mark Shuttleworth] это отменил. Ubuntu 16.10 предложил технический предпросмотр Unity 8, который, несмотря на ошибки и некие неровности

Отчасти жаль, что мы никогда не увидим Unity 8 в более отлаженной, конечной форме.

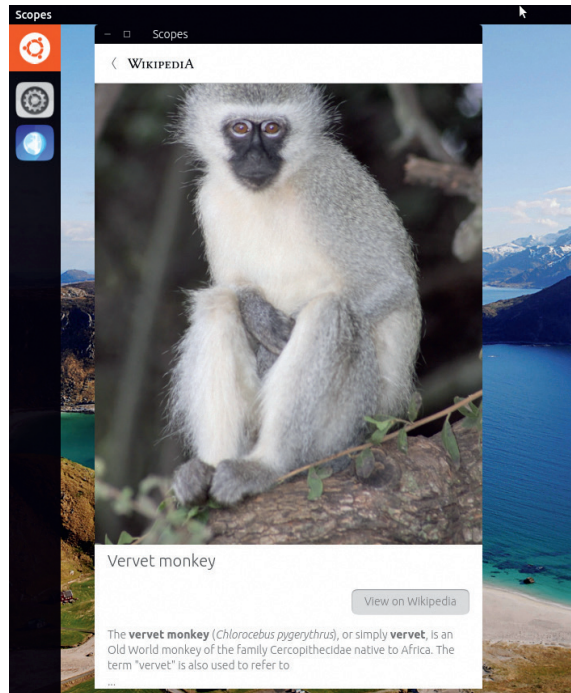


► Если перевести мышшь к правому краю экрана или провести по правой стороне, активируется этот стильный переключатель приложений, прямо как в Windows Vista году примерно в 2007-м.

по краям, всё же давал ясное представление о том, каким будет конечный продукт. В 17.04 мы по-прежнему имеем то, что по большей части является техническим предпросмотром, но таким, который предлагает куда более сложное взаимодействие и представление о том, куда всё могло бы пойти, если бы не силы рынка. Так чего же вам ожидать, если вы выберете эту стильную восьмерку? Что ж, давайте сразу устраним это препятствие; вероятно, вы бы хотели применять на рабочем столе совсем не то. Вас приветствует с виду пустынный рабочий стол с совершенно пустым окном Scopes (подробнее об этом через секунду) и немного странным (возможно, только для нас) смещением шрифтов. Но всё это мало чем отличается от Unity 7 — панель запуска находится слева, системная область (со знакомыми опциями громкости, сети и выхода) — вверху справа, и если вы со всем пылом кинетесь вправо, появится стильный переключатель приложений. Нажатие на кнопку Ubuntu (вверху слева) вызовет список приложений в лексикографическом порядке.

По поводу Scopes и их роли в будущем конвергенции Ubuntu было много шума. В сущности, эти приложения можно найти на других мобильных платформах, но при этом они по запросу могут включать настольный режим. Пустое окно Scopes можно заселить, нажав на стрелку вниз и сделав выбор из текущих предложений. К сожалению, выбор доступных Scopes довольно ограничен — есть несколько полезных/развлекательных, в том числе Reddit, Wikipedia и канал Weather. Выбрав свой перечень, вы сможете просматривать их, перетаскивая (или ударяя по ней, что кажется намного менее странным, если у вас сенсорный экран) верхнюю панель.

Отключите Desktop Mode [Настольный режим] в системном меню, и произойдет удивительная трансформация: активное приложение заполнит экран. Окно Scopes выступает в роли домашнего экрана на мобильных платформах, и поэтому его нельзя убить (по крайней мере, обычным оружием), поэтому в отсутствии остальных приложений оно заполнит экран. В этом смысле у scopes есть нечто общее с виджетами Android, поскольку они предлагают информацию в виде предпросмотра, и пользователь может ее расширить, открыв другое приложение. Например, Wikipedia score отображает подборку со страницы этого сайта «Основные статьи [Featured Articles]», а также строку поиска. Если выбрать статью, появится краткий обзор и одно изображение (при его наличии),



» Wikipedia scope, помимо всего прочего, очень полезен в поиске изображений животных, в честь которых названы релизы Ubuntu.

откуда пользователь сможет открыть браузер для просмотра остальной части.

Unity 8 впервые представили в Ubuntu 13.10, и тогда это было скорее эксперимент. В этом релизе мы увидели большой прогресс по сравнению с 16.10, но не станем рекомендовать ее для повседневного использования — в основном это по-прежнему технический предпросмотр (и теперь ни к чему не ведет). У нас было несколько сбоев и странных ошибок (но помните, что мы работаем с бета и пишем это в прошедшем времени), и в целом всё довольно шероховато. И всё же острые края окна смотрятся современно, новая системная область аккуратна, и в плане структуры придаться не к чему. Отчасти жаль, что мы никогда не увидим ее в более отлаженной, окончательной форме. Но зато мы, по крайней мере, можем рассчитывать увидеть в следующем году Ubuntu на GNOME. »

Заточение приложений в Libertine

Многие традиционные приложения на основе X11 не ладят с сервером отображения Mir, на котором работает Unity 8 — по крайней мере, не по умолчанию. Например, попробовав открыть Firefox или LibreOffice Writer, вы увидите всплывающий экран и вращающийся прогресс-индикатор, который никуда не приведет.

Эти приложения последнего поколения можно заставить работать с помощью среды песочницы Libertine. Она благополучно запускает старые пакеты в контейнере LXC.

Одна из (великого множества) причин подобного стремления уйти от X11 в том, что здесь нет встроенного механизма изоляции приложений друг от друга. Любой клиент может прочитать входящие события и данные окна от любого другого клиента, что сильно облегчает написание клавиатурных шпионов и прочих шпионских программ. Это серьезная проблема, особенно если учесть, что разные пользователи могут соединяться с одним и тем же X-сервером одновременно. С помощью Mir (и Wayland, другим сервером отображения будущего)

приложения изолируются друг от друга, и всё должно быть куда безопаснее.

Закрывая в песочницу старые приложения X11, их можно изолировать от остальной части системы и, посредством использования ряда контейнеров, друг от друга. Эти устаревшие приложения надо отдельно устанавливать в контейнер, даже если они уже существуют в вашей системе. Более того, каждый контейнер требует около 500 МБ (еще до того, как вы хоть что-то с ним сделаете), так что вы, возможно, решите использовать приложения X11 в Unity 7 как они есть. Некоторые пакеты вам придется установить заранее, до того, как вы увидите, как выглядит будущее, заставляющее работать прошлое:

```
$ sudo apt install libertine libertine-scope
```

Прокрутите список приложений, и вы найдете Libertine Manager. Запустите его, и вам предложат настроить «Classic Application Support». Затем вы увидите несколько опций: как назвать ваш контейнер, нужна ли ему 32-битная поддержка и надо ли вам защитить его паролем. Настроив всё это,

приступайте к установке пакетов, для чего выберите контейнер и нажмите на значок плюса наверху справа. Вы можете искать пакеты по имени или выбрать из недавно скачанных файлов .deb. Как мы и привыкли, можно администрировать контейнеры Libertine и заключенные в них пакеты из командной строки с помощью `libertine-container-manager`. Например, для установки всего пакета LibreOffice в контейнер с именем zesty можно ввести

```
$ libertine-container-manager install-package --id zesty --package libreoffice
```

Некоторые не-Mir-ные приложения будут работать таким образом без заключения в контейнер, через слой совместимости Xmir, но они собраны строками заглавия Unity 7 под теми, которые относятся к новому стилю. Это выглядит странно, но зато дает вам выбор, какую кнопку закрытия использовать. Стандартное приложение Ubuntu Terminal не запустится, но есть альтернатива Mir, которая с чрезмерной осторожностью попросит вас ввести пароль перед тем, как запуститься. Лучше перестраховаться, чем потом сожалеть.

Разновидности Zest

Нелюбовь к Unity не должна заставить вас отказаться от Ubuntu. Почему бы не попробовать другие официальные вкусы? Ам-ням-ням.

Критика рабочего стола Unity — дело обычное. Является ли причиной то, что она сместила пользовательские кнопки, или то, что кто-то не смог найти нужные кнопки, или просто те, кто сидит в Интернете, присоединились к подстрекателям (как они обычно и делают) — трудно сказать. Но какой бы ни была причина, нелюбовь к рабочему столу отнюдь не должна стать причиной отказа от всего дистрибутива. У Ubuntu много тысяч пользователей и тщательный процесс тестирования, который заботится об этих пользователях. Прилагаются огромные усилия, чтобы обеспечить работу приложений с самого начала, по умолчанию.

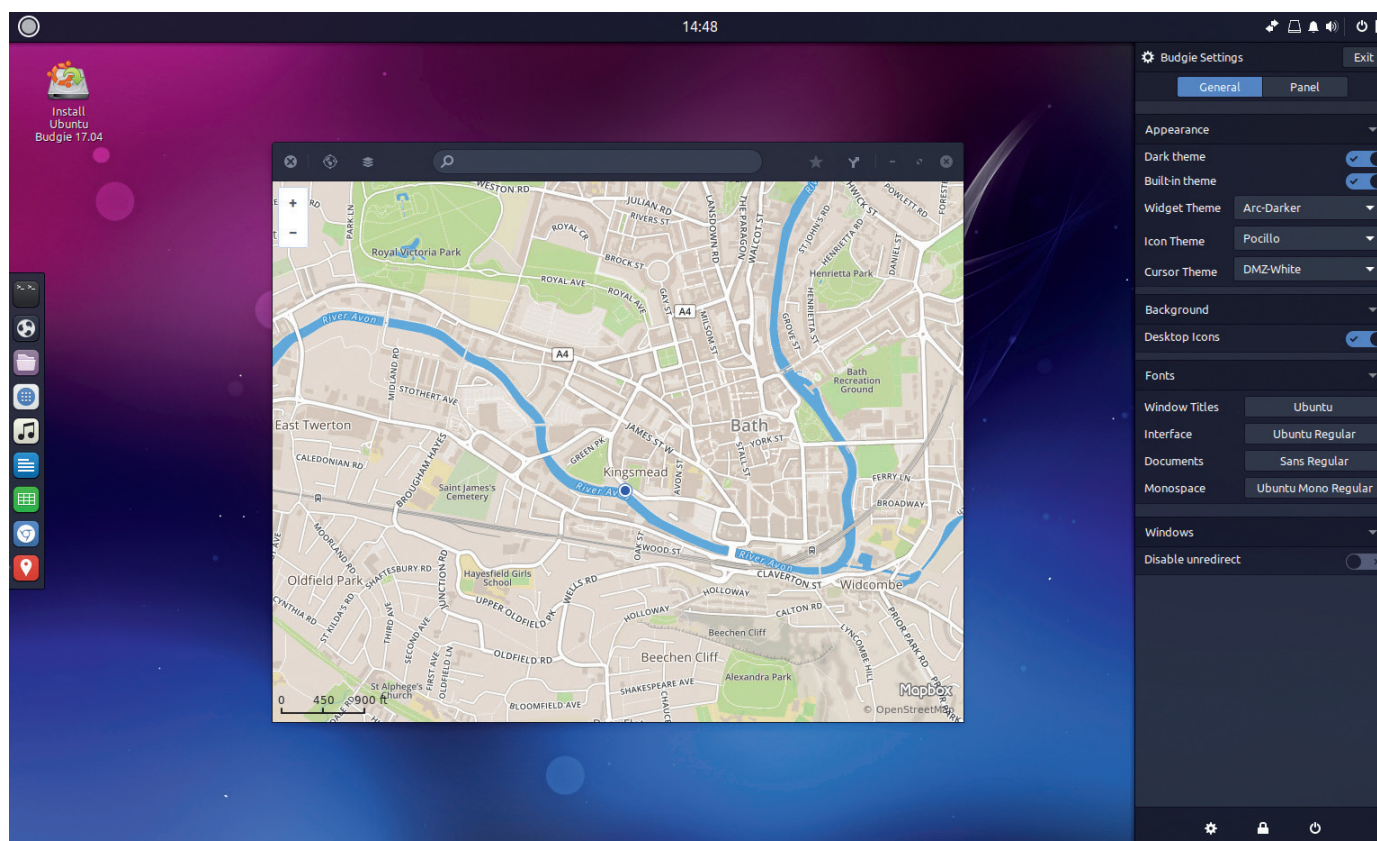
Конечно, есть и другие дистрибутивы, которые по-своему хороши, например, Arch Linux [Ред.: — Может, помолчим про Arch? Статья-то про Ubuntu!], однако слишком часто пользователи убегают по причине внешнего вида (они что, не смотрели *Красавицу и Чудовище?*). Итак, многие дистрибутивы ныне всё равно основаны на Ubuntu, так почему бы не выбрать официальную версию? Вы получите всю стабильность базы Ubuntu вместе со средой рабочего стола вашей мечты. Это если вы мечтаете о KDE, Xfce, LXDE, Mate, Gnome или Budgie.

Есть и другие официальные разновидности — Edubuntu (для школ), Ubuntu Studio (для художников, музыкантов, создателей видео и прочих творческих личностей, см. [LXF223](#)) и Ubuntu Kylin (для китайских пользователей) — но они всё равно используют Unity. В этом цикле релизов мы также скорбим по MythBuntu, который с 2005 г. облегал страдания, сопровождавшие настройку

MythTV. Увы, из-за сокращения команды разработки проект был свернут. Тем временем Lubuntu продолжает переход от устаревшего и полного ошибок *GTK2* к рабочему столу *LXQt*. В данном релизе смены караула не произойдет, но пользователи, жаждущие увидеть, как идут дела в легковесном, но современном рабочем столе на *Qt5*, могут для этого установить пакет *lubuntu-qt-desktop*. Как мы увидим позднее, эти и другие пакеты рабочего стола можно установить на любую разновидность Ubuntu, чтобы пользователи могли выбрать рабочий стол при входе в систему по своему настроению.

Здесь мы сосредоточимся на двух официальных версиях, Budgie и Gnome, потому что другие в этом релизе предлагают только небольшие обновления пакетов. Budgie — новейшее дополнение к семейству Ubuntu. Пернатый рабочий стол [Budgie — волнистый попугай, — прим. пер.] родился в дистрибутиве Evolve OS, ныне известном как Solus Linux (см. [LXF208](#)), однако с тех пор обзавелся внешней поддержкой. Budgie основан на Gnome, но из-за попыток создать рабочий стол с элементами, напоминающими о Gnome 2, хотя и с современными удобствами вроде *insta-search*, разработчики постоянно наталкиваются на необходимость обходных маневров. Всё это говорит о глубоких изменениях в экосистеме Gnome, где *GTK*, некогда общецелевой набор инструментов для создания привлекательных и стабильных виджетов, становится всё больше и больше зависим от Gnome. В результате попытки сделать что-то на *GTK3*, не используя рабочий стол Gnome,

➤ Меню настройки Budgie Raven удобно, и Gnome Mars тоже прошли долгий путь.

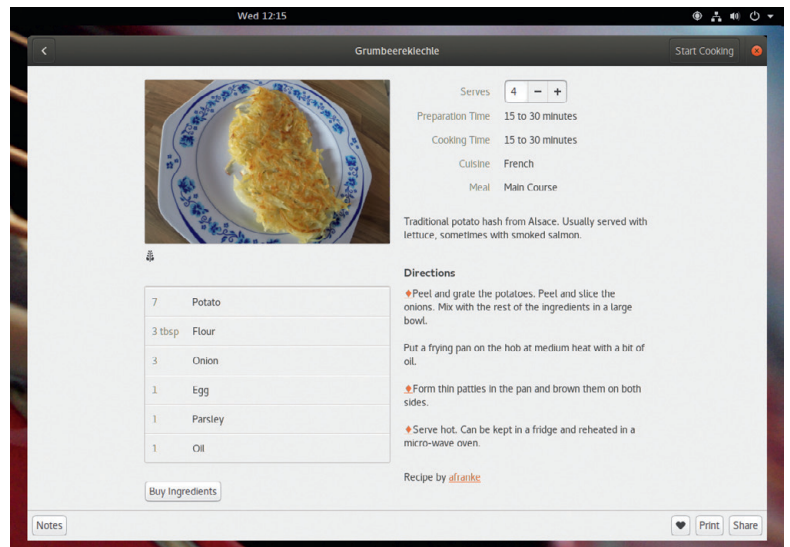


становится всё сложнее. Поэтому команда Budgie, как и LXDE, в будущем [Ред.: — Ха-ха] перейдет на Qt5. Интересно, что Unity 7 является альтернативной оболочкой для Gnome, которая реализована как плагин *Compiz*. До сих пор ей удавалось обходить эту ассимиляцию GTK-Gnome, частично благодаря многочисленным заплаткам, а частично благодаря использованию другого набора инструментов под названием Nux. Если всё это кажется вам слишком запутанным, не переживайте — так и есть, потому что Unity 8 и использует независимый от устройств Qt5/QML.

Вернемся к Budgie, и при первом же запуске вы увидите дружелюбный экран приветствия, где размещается весьма демократичный экран выбора браузера. Он напоминает тот, который ЕС заставил Microsoft реализовать в 2009 г., когда было принято решение о том, что использование *Internet Explorer* с Windows было антиконкурентным. По умолчанию стоит *Chromium*, но пользователи могут установить *Chrome*, *Firefox* или *Vivaldi* одним нажатием. Это приятная особенность, поскольку выбор браузера является постоянной проблемой среди пользователей — многие предпочитают *Google Chrome* и обижаются на то, что во многих дистрибутивах его трудно установить, и не меньшее количество удивляется тому, что этот образчик проприетарной мерзости установлен по умолчанию. Budgie, в отличие от своего родителя, рабочего стола Gnome, и (возможно, это необоснованно жестко) Unity, в первую очередь озабочен возможностью пользовательской настройки. Здесь имеется удобная боковая панель — *Raven Sidebar*, которая всплывает при щелчке по значку с краю справа на основной панели инструментов. Отсюда можно управлять виджетами, уведомлениями, структурой панелей и настройками шрифтов. На самом деле, просто поразительно, сколько настроек они умудрились сюда втиснуть без ощущения перегруженности. Впрочем, дорожная карта для следующего релиза Budgie показывает, что некоторые из этих настроек отобраны для включения в апплет основной конфигурации. Нас мучают некоторые сомнения по этому поводу: с одной стороны, хорошо иметь легкий доступ к этим настройкам, но с другой стороны, такие вещи, как размер шрифта, изменяются один раз (если вообще изменяются), что делает их доступность неактуальной.

Budgie использует область дока/программы запуска Plank, которая изначально размещена слева, но она меньше, чем в Unity, и меньше отвлекает. Ее можно настроить, чтобы она автоматически скрывалась или даже скрывалась по условиям [intellihide], и снабдить такими «доклетами», как часы, монитор CPU или настольный пейджер. Budgie также использует превосходное приложение *Terminix*. Оно не только умеет делиться горизонтально и вертикально (как *Terminator*), но также может сохранять вашу сессию в анализируемом .JSON. Как и в Unity, здесь довольно много от Gnome 3.24, но явно меньше, чем в Ubuntu Gnome, на который мы скоро обратим наш орлиный взор.

Этот релиз Ubuntu отлично соответствует релизу ядра 4.10, который вышел в большой мир за пять недель до выхода Zapus



» **Recipes** — новое дополнение к семейству приложений Gnome, а благодаря этим вкусным эльзасским картофельным оладьям оно очень приветствуется.

и, очевидно, был доступен для тестирования задолго до этого. Конечно, разные проекты следуют разной частоте релизов, и далеко не всегда всё так идеально совпадает. Один из таких примеров — рабочий стол Gnome, версия 3.24 которого вышла всего за две недели до Ubuntu. На момент написания она должна уже быть полностью инкорпорирована во все дистрибутивы Linux. Как бы то ни было, части Gnome 3.24 были использованы в базо-

Поразительно, сколько настроек умудрились втиснуть без ощущения перегруженности.

вом Ubuntu (Calendar, Videos, Disks). Во многом это благодаря графику выхода долгосрочных релизов GTK, который упрощает разработчикам вне Gnome задачу лучшего отслеживания библиотек и возможности избежать проблем, вызванных изменениями API. Однако от Gnome 3.24 можно получить больше (хоть и не всё), если использовать официальную версию Ubuntu Gnome или добавить рабочий стол (все 800 МБ) с помощью `sudo apt-get install ubuntu-gnome-desktop`. Ubuntu нужен густо залатанный *Nautilus* и терминал для интеграции с его особенностями, поэтому включены более старые версии (3.20). Подобным же образом *Gnome Software* (залатанный и переименованный *Ubuntu Software*) основан на редакции 3.22, но включает поддержку для Snaps и Flatpaks, чего не было в предыдущих Ubuntu.

»

Новинки Gnome 3.24

В самом последнем релизе Gnome не так уж много изменений, заметных пользователю, хотя в нем имеется *Night Light*, который настраивает температуру вашего дисплея для уменьшения количества синего света, который он рассеивает ночью. Если дисплей в ночное время теплее, то снижается напряжение глаз и усталость.

Подобные программы существуют уже некоторое время — например, имеется кросс-платформенный, но с закрытым кодом *f.lux*, или *Redshift* с открытым кодом. Люди (в частности, авторы

статей для журналов о Linux) проводят всё больше времени, пляясь в монитор по ночам, от чего страдают и здоровье их глаз, и их суточные биоритмы. Поэтому очень приятно, что эта функция теперь входит в Gnome по умолчанию, теперь это один из не столь неподходящих значков в системном лотке. Впрочем, в данный момент *Redshift* не работает с *Wayland* и, похоже, не согласуется с солидными приложениями типа *Chromium*, в то время как *Night Light* во время наших тестов отлично себя проявила со всеми.

Помимо защиты наших глаз, Gnome 3.24 включает несколько новых приложений, которые можно установить отдельно — вы, конечно, ожидали *Recipes*. Оно умеет составлять для вас список покупок и позволяет вам делиться собственными рецептами. Здесь есть и полноэкранный режим, что бывает удобно, если у вас сенсорный ноутбук на кухне. Также здесь есть *Games*, способное управлять не только встроенными играми без DRM, но также вашей библиотекой Steam и эмуляторами на *libretro*.

Графические драйверы

Драйверы с открытым кодом далеко продвинулись, но матерые геймеры с новыми картами Nvidia захотят себе проприетарные версии.

Многие используют Ubuntu в качестве игровой платформы, и у них есть на то причины — помимо SteamOS, именно здесь проходит большая часть тестирования в Linux. Фактически, среда исполнения Steam (устаревающая) основана на старых библиотеках Ubuntu. Установить Steam на Ubuntu очень легко — просто активируйте репозиторий Multiverse из приложения Software & Updates (введите 'soft' в Dash, и вы его найдете), и он будет доступен из приложения Ubuntu Software или через старую добрую

```
$ sudo apt-get install steam
```

Пакет, который устанавливается таким образом — это всего лишь своего рода начальный установщик, который скачивает и устанавливает клиент Steam в вашу домашнюю директорию. Таким образом Steam заботится о своих обновлениях вне менеджера пакетов, и по большей части отлично справляется с задачей не повредить себя. Если вы собираетесь играть в игры с хорошей графикой, или даже игры со средней графикой, и у вас приличная видеокарта, то вы, вероятно, решите установить проприетарные графические драйверы. Точнее, если у вас карта Nvidia, то вы,

видимо, именно так и решите, а если более новая карта AMD, то, наверное, нет. Благодаря новой среде драйверов AMDGPU поддержка с открытым кодом более новых (через минуту мы это проясним) карт AMD на самом деле очень хорошая. Имеется проприетарный драйвер, AMDGPU PRO, но он предназначен для более профессиональной работы OpenCL, а не для повышения производительности игр. Изначально новый драйвер поддерживал только "Volcanic Islands" и более поздние карты (серию Rx 200, появившуюся в 2013 г., и более новые), но с ядром 4.10 поддержка вернулась к картам Southern и Sea Islands (HD7000 и HD8000). Более старые проприетарные драйверы ATI, известные как Fglrx, больше не поддерживаются, и не работают с более новыми версиями X, так что если у вас нет мазохистской тяги к старому оборудованию, их лучше избегать. Купите новую видеокарту или используйте старый свободный драйвер radeon.

Картам, в которых используется архитектура Nvidia Maxwell (GTX900) и Pascal (GTX1000), действительно нужен проприетарный драйвер, чтобы задействовать весь их далеко не ничтожный игровой потенциал. Драйвер можно установить из вкладки Additional Drivers [Добавочные драйверы] в утилите Software & Updates. Но поскольку они обновляются не так часто, как Nvidia выпускает новые драйверы, вы, возможно, решите использовать новый PPA графических драйверов. Уже не раз доверчивые пользователи попадали в переплет из-за того, что скачивали бинарники драйверов прямо с сайта Nvidia, использовали комбинацию экспериментальных PPA, копировали и вставляли инструкции с Reddit, или делали всё вышперечисленное. PPA графических драйверов задуман как универсальный источник для тех, кто готов рискнуть и получить чуть более нестабильные драйверы, но за счет более новых функций. Хотя драйверы из этого PPA способны иногда демонстрировать странное поведение или сбои, они не должны привести к тому, что оборудование откажется грузиться [Пед.: — Безопасная загрузка вам в помощь] или к потере данных.

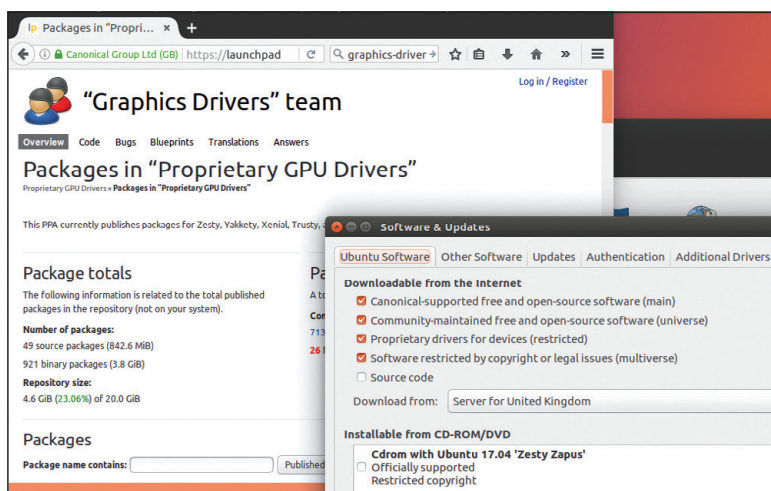
Чтобы использовать более новые драйверы, вам нужно лишь

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

```
sudo apt-get update
```

```
sudo apt-get install nvidia-378
```

Естественно, в репозиториях полно игр FOSS, если проприетарные программы вызывают у вас гадливую дрожь.



➤ Проприетарные драйверы можно установить из вкладки Additional Drivers, но если вам нужно что-то поновее, загляните в PPA.

Nouveau

Nouveau, драйвер с открытым кодом от Nvidia, является замечательным проектом. В отличие от AMD, Nvidia внесла очень незначительный вклад в развитие свободного кода для своих карт. Раньше был свободный драйвер nv, который содержал код настолько запутанный, что многие разработчики лишились рассудка, просто пытаясь его проанализировать (шутка).

Итак, проект Nouveau был запущен в 2005 г. с целью обратного инжиниринга оборудования и написания драйверов, которые могли бы быть

свободными, иметь поддержку и не доводить до безумия. Nouveau — драйвер по умолчанию для оборудования Nvidia и обычно работает хорошо, то есть пользователи после загрузки не обнаруживают перед собой черный экран. Однако если вам необходима производительность, Nouveau вам вряд ли поможет. Причина в том, что эти карты загружаются в состоянии низкого энергообеспечения, и без использования экспериментального (или проприетарного) кода их крайне трудно вывести из этого состояния.

Работа в этой области ведется, и ядро 4.10 добавляет ряд функций разгона (правда, вручную) для карт на основе Kepler (серия GTX600-700). Более старые карты не подходят для современных игр, поэтому их владельцы спокойно могут использовать свободный драйвер (хотя устаревшие проприетарные драйверы серии 304 поддерживают некоторые из них). Те, кто интересуется экспериментами с новыми функциями Nouveau, должны будут как следует разобраться с опцией `nouveau.config=NvBoost=2` ядра и пунктом `/sys/kernel/debug/dri/0/pstate`.

Умеренность — это хорошо

Помимо новости о конце Unity, данному релизу нечем особо привлечь внимание. Объясним, почему это хорошо.

Вот вам вводка: все эти громкие слова на обложке несколько неверны. В данном релизе нет особо революционных функций; совсем не факт, что он стабильнее предыдущего; и будем реалистами — обновив LTS до этого релиза, вряд ли вы приобретете что-то сверх большого номера версии. [Ред.: — За молчки уже!]

Хотите — верьте, хотите — нет, но всё это Хорошо: сильные изменения огорчают людей, как и установка новой операционной системы каждые полгода. На самом деле, если вас устраивает ваш Ubuntu 16.04, последуйте совету Canonical и сохраните его — LTS очень хорошо тестируются и поддерживаются в течение пяти лет, в отличие от промежуточных релизов, которые поддерживаются только до выхода нового. Этот релиз, как и многие до него, очень аккуратно вносят улучшения в проверенный рецепт.

Да, за этот период были большие изменения в Ubuntu — мы видели, как приходят (*systemd*) и уходят (*Upstart*) системы *init*; мы видели, как меню приложений перемещаются из окна приложений на верхнюю панель (и затем опционально в строку заголовка); и мы не проливали слез по гадкому *Ubuntu Software Centre*. Приложено немало усилий для поддержания совместимости с более новыми версиями компонентов *Gnome*, на которые опирается *Unity*, и, говоря в общем, для отслеживания новых релизов всего, что есть в репозиториях. Но по большей части любой, кто знаком с Ubuntu примерно с релиза 12.04, не должен затрудняться с навигацией по следующим Ubuntu, и это относится также и к 17.04.

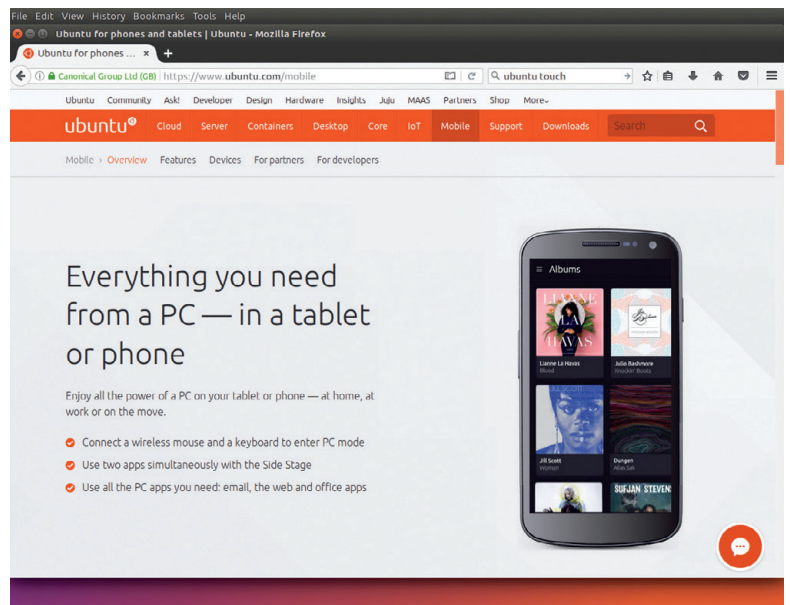
За исключением *Windows 8*, мы могли бы сказать то же самое о *Windows* с *Windows 95*, и об ОС, известной ранее как *OS X*, с момента ее появления в 2001 г. Инновации — это, конечно, здорово, но хорошо бы также не ломать то, что пока не сломано; ключевая характеристика любого хорошего рабочего стола заключается в том, что он не путается у вас под ногами, и представление о блестящих, прорывных функциях существует совершенно независимо от этого.

Добавьте сюда общую статистику, говорящую о том, что люди всё меньше и меньше занимаются своими рабочими столами (согласно недавним исследованиям, сегодня *Android* более популярен для работы в сети, чем *Windows*), предпочитая использовать приложения на своих телефонах, планшетах или глазных имплантах, и вы начнете понимать, что попытки привлечь пользователей к использованию Ubuntu на рабочем столе с помощью кричащих заголовков обречены на провал. Подобный подход только разозлит или отпугнет имеющихся пользователей. И, говоря относительно, их никогда не было особо много, для начала.

Вне рабочего стола

Не забывайте, что Canonical — это компания, и настольный Ubuntu делает очень мало, чтобы увеличить ее прибыль. Прибыль получают благодаря выгодным корпоративным контрактам на поддержку и предложений облачных услуг. Версия *Server OS* от Canonical и, в частности, множество ее индивидуально настроенных облачных образов имеет на порядок больше пользователей и установок, чем настольный Ubuntu, и со временем это соотношение будет только расти.

Недавно на первый план вышла *Ubuntu Core*, уже применяемая для работы на дронах, роботах, базовых станциях *SDR*, и т.д.



► Когда наконец-то выйдут *Unity 8* и *Ubuntu Phone*, все начнут жаловаться на... Пойдите, как? Говорите, этого не случится? О Боже.

Она питает надежду (через транзакционные обновления, *AppArmor* и фокусы с контейнерами) принести некое благообразие в дикий и небезопасный мир Интернета Вещей, и благодаря разнообразным промышленным партнерствам Canonical скоро будет гораздо больше устройств, работающих на этой ОС.

У всех этих вариаций по-прежнему солидная доля общего кода, что помогает избежать чрезмерной фрагментированности процесса разработки и снимает некую часть бремени поддержки свобод-

Инновации — это здорово, но хорошо бы не ломать то, что пока не сломано.

ной ОС рабочего стола. Универсальный и безопасный механизм пакетирования, например, *Snaps*, относится к установкам на рабочем столе и на устройствах. Цель уже оставленного перехода на *Unity 8* заключалась в унификации интерфейсов пользователя на настольных и мобильных системах, опять же подтверждая идею общей, конвертируемой базы кода.

Совершенно другой подход в *Red Hat*, который использует свободный дистрибутив *Fedora* в качестве области технологической подготовки для коммерческого *Red Hat Enterprise Linux*; и совершенно другой подход в *SUSE*, «другом» корпоративном игроке *Linux*, который спонсирует *openSUSE*. Но чтобы делать деньги на свободном ПО, нужны инновации и изобретательность. Некоторые пользователи могут не согласиться с точкой зрения Canonical, и прелесть открытого кода в том, что найдутся дистрибутивы и для этих пользователей; но с нашей точки зрения, это отличное время для Ubuntu и для Linux в целом. LXF



Maker Faire UK 2017

Лес Паундер снова посетил Ньюкасл-на-Тайне — не ради любования его туманами, а чтобы увидеть новейшие проекты мастеров-любителей.

Maker Faire UK [Ярмарка Умельцев] — это крупнейшая в Великобритании выставка устройств, сделанных своими руками; толпы людей съезжаются сюда со всего мира, чтобы принять участие в этом недельном празднике мастеров-любителей. Разнообразие — главная тема каждой Maker Faire. Вы можете подумать, что это всё про схемы и код; но взглядевшись глубже, вы обнаружите множество проектов, которые служат другим целям. Гаражные мастера печатают на 3D-принтерах индивидуальные протезы для детей-инвалидов; музыканты делают совершенно потусторонние инструменты из всевозможных шуток, задуманных отнюдь не для

извлечения музыки; а детишки на глазах вырастают в новых умельцев.

На каждом Maker Faire UK среди участников встречаются громкие имена, и в этом году, среди прочей публики, мы увидели здесь Raspberry Pi Foundation, CPC и Kitronik. Независимо от возраста и квалификации, здесь все участники были «на равных»: ведь во всех нас живет неистребимое стремление узнать, как устроены вещи и как мы сможем использовать их в нашем следующем проекте. Кстати, видеть столько малолеток, представляющих собственные проекты и часенько демонстрирующих старшим поколениям, как реализовать отличную идею — тот еще удар по самолюбию.

Судя по экспонатам, более всего мастеров вдохновляют всё те же две платы: Raspberry Pi и Arduino. На базе Raspberry Pi были роботы, следящие камеры и даже световая скульптура, изображающая сердцебиение участников мероприятия. Arduino управляет 3D-принтерами с открытым исходным кодом и лазерными резаками. Эти платы вошли в сообщество мастеров-любителей, и Ярмарки Умельцев являются отражением общего интереса — с каждым годом мы видим, что их применение становится все более разнообразным.

Участие в Maker Faire UK — это шанс опробовать новинки и поговорить с людьми, которые их создали. Каждый делится своими идеями и опытом. А это и значит быть мастером-любителем.

Робин Хартли [Robin Hartley]



Робин — новое лицо в сообществе «самоделкин». Он приехал на Maker Faire UK, чтобы показать свой первый проект: клавиатуру, программируемую на выполнение нескольких задач одним нажатием кнопки. Причем особенность его проекта в том, что сделать это можно с помощью блочного интерфейса.

Linux Format: Привет, Робин, спасибо, что нашли время поговорить с Linux Format. Расскажите нашим читателям о себе.

Робин Хартли: Привет, я Робин, учусь в магистратуре Шеффилдского университета, по специальности «Химическое машиностроение».

LXF: Значит, вы не инженер-электронщик?

PX: Нет, электроника и «изобретательство» — это просто хобби, потому что мне это действительно нравится. По ходу я научился проектированию печатных плат, web-разработке и 3D-дизайну CAD.

LXF: И давно ли вы начали всё это изучать?

PX: Программировать на Arduino я начал около двух лет назад, потому что на одном собеседовании сорвал, что умею. К счастью, у меня было шесть месяцев, чтобы научиться и подготовиться.

LXF: В эти выходные на Maker Faire UK вы представите на стенде свое последнее изобретение. Можете ли вы рассказать нам об этом подробнее?

PX: Название у него довольно забавное — Amazing Shortcut Keypad [Чудо-клавиатура быстрого доступа]. Компьютеры умеют делать невероятные вещи, сегодня у нас есть такие технологии, которые всего 20 лет назад были просто мечтой. Но за все эти годы интерфейс практически не изменился. Всё та же клавиатура, всё та же мышь. А как обычно строится взаимодействие с новой функциональностью? Через громоздкие меню и необходимость перемещаться по ним при помощи мыши или с помощью типизированных команд, используя сочетания клавиш. И то, и другое, не особо эффективно.

Keypad я изначально создавал для себя, потому что в тех программах CAD, что я использовал, надо было запоминать и вводить серии команд, и мне это надоело. Я просто хотел нажать кнопку, чтобы команды печатались за меня. И теперь я могу набирать команды одним нажатием кнопки. Все это стало возможным благодаря тому, что в основе моего проекта лежит Arduino, который работает как клавиатура, а для программирования Keypad я создал простой редактор блоков, позволяющий каждому настроить выполнение любых действий или назначить клавиши быстрого доступа по своему вкусу. Я также хочу, чтобы пользователи могли делиться своими картами, чтобы помогать другим членам сообщества. Например, пользователь *Photoshop* может поделиться своим макетом, чтобы и другие люди использовали *Photoshop* более эффективно.



LXF: Такая клавиатура может применяться очень широко, не только потому, что она более эффективна, но и потому, что ее можно использовать в специальных сферах медицинских/вспомогательных технологий.

PX: Да, с ней можно много чего придумать, а другие могут подать идеи, как еще больше расширить ее потенциал.

LXF: Почему вы вдруг решили стать мастером-любителем? Откуда эта потребность «понять, как всё это работает»?

PX: Я считаю, что главная «фишка» всех любителей изобретать — не в том, что мы хотим что-то создать, а то, что мы хотим увидеть, как всё оживает. Независимо от того, что мастера намерены сде-

лать, можно пообщаться с публикой — выслушать их рассказы, вдохновиться их удивительными проектами и, возможно, завязать знакомства. Вдохновение и азарт — это то, что претворяет в жизнь проекты, а участники и широкая публика в Maker Faire UK здорово вдохновляют.

LXF: На Maker Faire UK мы видим всю широту любительских проектов: от устройств для помощи инвалидам до обучения вязанию и шитью. Есть ли какие-то границы у того, чем может и чем не может заниматься «мастер-любитель»?

PX: Это сложный вопрос! Границу провести не получится: ведь нельзя же сказать, например, что лазерной резкой — может, а 3D-печатью — нет. Это больше касается отношения к делу, готовно-

О СООБЩЕСТВЕ

Сообщество играет огромную роль в жизни движения. Все помогают друг другу идеями и советами.

лать, у каждого свои особые навыки и горячее желание учиться и добиваться цели. Это характеризует мастера гораздо больше, чем наличие инструментов и знание, как ими пользоваться.

LXF: То есть можно сказать, что это такой склад ума?

PX: Да, это почти как в стартапе, там это называется «установка на рост» — то есть способность учиться, развиваться и постоянно становиться лучше. То же можно применить и к любителям, чтобы сформировать аналогичный тип мышления у мастеров-любителей.

LXF: То есть Maker Faire UK — это гораздо больше, чем просто крутые технологии?

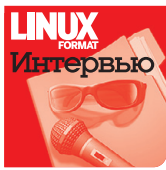
PX: Действительно, я на своем стенде буду совсем один, поэтому у меня не будет времени походить и посмотреть, что выставляют другие, но здорово,

сти учиться, желания претворить что-то в жизнь и чувства общности. Сообщество играет огромную роль в жизни этого движения, все помогают друг другу, предлагают советы и варианты развития.

LXF: Пока ваша клавиатура все еще в стадии прототипа, вы не думаете собирать средства на ее производство?

PX: Да, мой сайт — theamazingshortcutkeypad.com, и планируется начать сбор средств в середине июня 2017 г. Мне очень интересно перейти от того, что я спаял в своей комнате, в такое, во что у людей появится желание вкладывать деньги; довести технологию до состояния, готового к продажам, и параллельно приступить ко всем формальностям (прохождения через нормативные акты). Нужно быть готовым к этому познавательному путешествию, ведь это естественное развитие любительских проектов.

Эдриан Мак-Юэн [Adrian McEwen]



Эдриана в сообществе знают хорошо, он даже написал книгу о разработке проектов Интернета вещей. Он сторонник независимого производства и распространения любительских продуктов в коммерческой среде.

LXF: Привет, Эдриан, спасибо, что нашли время поговорить с LXF, можете ли вы рассказать читателям немного больше о себе?

Эдриан Мак-Юэн: Обычно я говорю, что я — тот человек, который подцепляет к Интернету всякие странные штуковины. То есть машину, испускающую мыльные пузыри, подключаю к Twitter или какие-нибудь датчики, измеряющие энергию волн, размещаю на Ла-Манше и отправляю эти данные по 3G для анализа. Я занимаюсь самыми разными и интересными делами. И еще я написал *Designing the Internet of Things* [Проектирование Интернета вещей] для издательства О’Рейли. Наконец, я один из основателей хакерспейса в Ливерпуле, известного как DoES Liverpool.

LXF: И давно вы занимаетесь любительскими проектами?

ЭМ: Я бы сказал, по крайней мере лет десять, хотя своего рода мастером-любителем я был всегда. Но в 2006/2007 г. я впервые услышал об Arduino. Фактически, в 2007 г. я купил свою первую плату этой марки на заводе Arduino, когда жил в Турине.

LXF: То есть можно сказать, что Arduino стал вашим проводником на этой стезе?

ЭМ: Да, Arduino явно этому помог. Я работал над встроенным программным обеспечением в течение всей моей карьеры, и это всегда были проекты, связанные с Интернетом. Например, я был



LXF: Вы не только мастер-любитель, но и сторонник независимого производства.

ЭМ: Да, с 2002 г. у меня своя компания, мы создаем под заказ проекты Интернета вещей. Гаражное производство означает, что у меня есть доступ к лазерным резакам, 3D-принтерам и множеству инструментов, а также то, что я могу сам изготовить почти всё, что мне нужно. Но компания всегда была заинтересована в индивидуальном подходе к разра-

У нас есть люди, которые приходят в DoES Liverpool с тем, что они хотят сделать на 3D-принтере на продажу... сейчас в мейкерспейсе порядка 200 таких объектов, но чтобы произвести больше, нужно понимать, как устроена местная система снабжения. Так как все больше людей занимается проектами в DoES Liverpool, мейкерспейс дает вам такую возможность, поскольку это реальное сообщество производителей, которые делятся своей информацией. Так что ваш путь в производство лежит через мейкерспейс, где вы получаете необходимые инструменты, а затем узнаете о замечательном сообществе мастеров-любителей и производителей, которое поможет вам реализовать ваш проект.

LXF: Для желающих стать мастером-любителем: есть ли какие-то шаги, которым они должны следовать?

ЭМ: Нет, в это сообщество попадают совершенно по-разному, и у каждого своя история о том, что его сюда привело. Для меня это был проект, который я хотел осуществить, именно он заставил меня пойти учиться, чтобы узнать нечто новое и ценное для себя. Гаражные мастера — очень приятная и пестрая публика: люди, сгорбившиеся за ПК — это лишь стереотип, реальное сообщество объединяет представителей самых разных сфер — искусство, изобразительное искусство, инженерное дело, высокая мода, устойчивый интерес к технологии помогает делать его еще разнообразнее; и тот стереотип, что сообщество в основном маскулинно, тоже не оправдывается.

О НЕЗАВИСИМОМ ПРОИЗВОДСТВЕ

Я не хочу проходить через всю систему венчурного капитала и ездить в Китай ради производства.

частью команды, которая в 1997 г. сделала первый web-браузер для мобильного телефона. В 2000-х я попробовал несколько других одноплатных компьютеров, таких как Sun Spots, но все они были довольно дорогими и сложными в использовании. И тут появился Arduino, всего за £20, не требующий специального программного обеспечения или знаний, чтобы начать его использовать. Таким образом, если бы Arduino мне не понравился, £20 были бы моей максимальной потерей, и эта плата просто пылилась бы на полке. Именно цена и легкость использования стали решающим фактором, подтолкнувшим меня к тому, чтобы попробовать себя в электронике.

ботке и производству продукта. Например, мы делали звонок Ackers Bell с доступом к Интернету, который можно связать с интернет-магазинами, чтобы слышать, когда в магазине совершается продажа. Поэтому относительно продукта мы подумали: «Как бы нам побольше сделать самим?» — я не хочу проходить через всю систему венчурного капитала, добывать кучу денег и ездить в Китай ради производства 20000 штук, которые мне же потом и продавать. Меня интересовало, как увеличить объемы и привлечь больше мастеров-любителей. Расширить производство, но без подобного риска — это действительно непростая задача, а у makerspace есть такие связи в производстве.

Лоррейн Андервуд [Lorraine Underwood]



Лоррейн — из Ирландии, и она работает с учителями в Ланкашире, помогая им обрести уверенность в преподавании информатики. Лоррейн является новичком

в сообществе мастеров-любителей, и это ее первая Maker Faire UK, где ей предстоит представить новый проект: Nora the Explorer, робот виртуальной реальности от первого лица.

LXF: Привет, Лоррейн. Не могли бы вы рассказать читателям немного больше о себе?

Лоррейн Андервуд: Я координатор от Computing at School в Ланкастерском университете. Computing at School — это финансируемая правительством организация, которая помогает учителям преподавать информатику. Сегодня на Maker Faire UK я приехала со своим проектом Nora the Explorer.

LXF: И что же такое Nora Explorer?

Л.А.: Nora — это робот-машинка на базе Raspberry Pi Zero. Сверху на ней расположена камера, управляется она с помощью Wiimote, а передача видео с камеры осуществляется на гарнитуру виртуальной реальности, позволяющую игроку видеть всё от первого лица.

LXF: Поразительно, то есть Raspberry Pi Zero W может работать с видео, Bluetooth и Wi-Fi во время движения?

Л.А.: Да, Wiimote подключается через Bluetooth, а Wi-Fi в Zero W генерирует точку доступа, к которой можно подключиться с помощью гарнитуры мобильного телефона. Принцип тот же, что и в дронах FPV.

LXF: И что вдохновило вас на создание этого?

Л.А.: Просто так, захотелось попробовать свои силы, ну и в Maker Faire UK тоже хотелось поучаствовать. Я, можно сказать, новичок в среде мастеров-любителей и пробую себя в разных жанрах. В прошлом году я побывала на Ливерпульском Makefest, и была просто в восторге от этого сообщества. Но на Maker Faire UK мне хотелось стать уже участником, так что мне нужна была классная идея. Я видела, как кто-то делает похожий проект с Arduino, передавая видео пользователю как радиосигнал, но тот проект был слишком затратный. А я на всё смотрю с позиции: «Я могу сделать это дешевле!» — что и привело к созданию Норы.

LXF: Вот так и работает мышление мастера-любителя: «Я могу это сделать!»

Л.А.: Да! И я всегда добавляю: «причем дешевле!».

LXF: Так насколько вы новичок в этом сообществе?

Л.А.: В прошлом году я принимала участие в Picademy, обучающей программе для преподавателей



от Raspberry Pi Foundation. Имею ученую степень по информатике, и всегда любила программировать. Но я никогда не занималась вычислениями в физическом аспекте, в аспекте создания чего-то. А в Picademy у меня было время. На самом деле, большую часть курса у меня было ощущение: «не, это не мое», поскольку я не любитель *Minecraft*, и слуха у меня нет; но когда я научилась зажигать

один и тот же источник питания. Авось, в следующем проекте я все-таки сломаю Pi!

LXF: И каков будет ваш следующий шаг?

Л.А.: Следующий большой шаг — изучение электроники. Понятия не имею, как это сделать. Можно ли добиться этого самому? У меня есть книга по электронике, через которую я медленно проди-

О NORA THE EXPLORER

Это робот-машинка на Raspberry Pi, передающая видеосигнал клавиатуре виртуальной реальности.

простой светодиод, это было удивительно и пробудило мой интерес. С тех пор я стала делать проекты с Neopixels, используя Raspberry Pi 3 и лестницу в своем доме! Температура на улице отображалась с помощью строки Neopixels внизу моей лестницы. Данные брались из общедоступного API, который предоставляет местный прогноз погоды, определяя ваше местоположение на карте. Код просто фиксирует температуру, а затем использует условный тест, запускающий должные неопиксели. Если температура поднимается выше 25 градусов, лестница загорается красным светом, но, живя в Великобритании, я вряд ли это увижу! С кодом мне было просто, поскольку я уже программист, а с электроникой пришлось потрудиться. Но, к счастью, мне помог мой муж, он отлично разбирается в электронике. Я немного боялась сломать Pi. Например, я не знала, что Neopixels и Pi могут использовать

один и тот же источник питания. Авось, в следующем проекте я все-таки сломаю Pi!

LXF: У вас было время посетить другие стенды Maker Faire UK?

Л.А.: Да, я с удовольствием прошлась по выставке и поговорила с участниками. Я познакомилась с Джонатаном Сандерсоном [Jonathan Sanderson], автором потрясающего проекта «Heart of Maker Faire [Сердце Maker Faire]»: он записывает ваше сердцебиение, а затем проигрывает запись, используя множество Neopixels. До этого мы с ним обсуждали в Twitter проблемы, с которыми сталкивались... Maker Faire — отличное место, чтобы наяву увидеть тех, с кем общаешься в Twitter, что-то с ними обсудить и поделиться своими идеями. **LXF**



Тестируем Linux

Значит, вы думаете, что ваша система быстра? Быстрее, чем у **Джонни Бидвелла**? Почти наверняка. И теперь вы можете доказать это с помощью нашего прекрасного руководства по тестированию скорости.

Различные аппаратные компоненты в вашем компьютере работают с определенной скоростью или имеют легко достижимые пределы скорости. Если ваш жесткий диск или SSD подключен к шине SATA 3.0, то его теоретический максимум скорости передачи — 600 МБ/с, тогда как чудесный m.2 SSD (подключенный к достаточному быстрому слоту PCIe) в удачные дни легко достигнет 2,5 ГБ/с, а сама шина (использующая 4 базовых двунаправленных шины PCIe 3.0) может выжать 3,9 ГБ/с.

Да, ваш CPU постоянно изменяет частоту в соответствии с нагрузкой (как и ваш GPU), и да, это можно разогнать, но все эти числа можно проследить или рассчитать. Проблема в том, что большую часть времени они не соотносятся с реальной производительностью, ибо реальные операции

затрагивают множество разных аспектов системы. Например, даже при всех данных о конфигурации всего задействованного оборудования, трудно сказать, насколько быстро система загрузит базовую установку самой последней версии Fedora. Точно так же вы не сумеете просчитать, какой FPS

Реальные измерения хитры и включают все возможные нюансы.

вы увидите, выкрутив все до 11 в *Shadow of Mordor*. Реальные измерения весьма хитры, поскольку включают все возможные нюансы — издержки файловой системы; пути кода, используемого графическим драйвером; задержки, вызванные

планировщиком ядра. Сложно предсказать, но, за исключением нескольких моментов, не так уж сложно измерить.

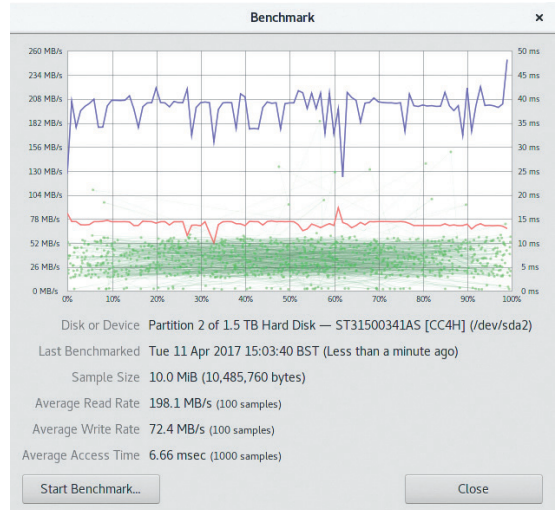
Сравнительные тесты — это черная магия, которая позволяет выполнять такие измерения. Она включает запуск стандартных программ, сравниваемых на разных компьютерах; каждая программа тестирует определенный аспект системы. Но ничто никогда не бывает слишком просто, и в сравнительном тесте легко ошибиться. Могут вмешаться фоновые программы, термозащита, экономия заряда ноутбука и композитинг дисплея. В частности, игры имеют тенденцию работать лучше на одном производителе процессора или драйвере GPU. Использовать определенный бренд и считать, что результаты покажут объективный рейтинг, весьма неумно.

Несмотря на всю противоречивость мнений о *systemd*, эта система инициализации дает простой и однократный способ измерения времени загрузки: запуск *systemd-analyze critical-chain* выдаст хорошее измерение периода ожидания от *Grub* до экрана входа в систему. Больше подробностей вы получите с помощью *systemd-analyze blame*, которая покажет, сколько занимает процесс завершения каждого отдельного сервиса. Помните, что сервисы запускаются параллельно, и долго работающие задачи (скажем, обновление базы данных *locate*) происходят в фоне, так что они не путаются под ногами, как некоторые могут заподозрить. Крутая функция профиля загрузки *systemd* — его способность создавать изображения: например, *systemd-analyze plot > ~/boot.svg* графически отобразит данные от упомянутой команды *blame*, создав файл в вашей домашней директории, который можно просмотреть в *web-браузере* или программе векторной графики (например, *Inkscape*). Наконец, более подробный график можно создать с помощью *systemd-bootchart*. Это немного сложнее и требует изменения *grub.cfg*, и если вы не в курсе структуры этого файла, то, вероятно, решите пропустить эту часть. Некоторые дистрибутивы (например, *Ubuntu* и *Fedora*) выделили эту функцию в отдельный пакет, *systemd-bootchart*, который надо устанавливать. Модуль *bootchart* выводится из параметра ядра *init*, обычно исполь-

сей оказываются одна поверх другой), но зато дает больше информации, чем график *systemd-analyze plot*, например, по потреблению CPU или вводу/выводу с диска. Оптимизацией процесса загрузки по большей части уже никто не занимается — *systemd* запускает части параллельно и знает, какая часть от какой зависит, так что вряд ли вы чего-то добьетесь, перетряхивая их. В общем, если вы хотите, чтобы ваша система грузилась быстрее, вероятно, лучше всего будет заменить этот старый жесткий диск с вашей ОС на сияющий новенький SSD.

Радар для диска

А раз уж вы сделали это, почему бы не провести сравнительный тест вашего нового приобретения? Здесь вам поможет скромная утилита *dd* (та самая, которую вы используете для записи ISO Linux на USB-брелок) с поправкой на пару ошибок. Метод, демонстрируемый нами, записывает в обычный файл, поэтому зависит от файловой системы и структуры раздела, но должен (если только не происходит какого-то скрытого кэширования за счет оборудования) предоставлять точные данные о скорости последовательного чтения и запи-



► **Disks сообщает:** наша скорость записи в половину меньше, чем скорость чтения, что вполне в порядке вещей.

тестовую машину, не особенно хорошую, *dd* сообщает следующее:

```
1048576000 bytes (1.0 GB, 1000 MiB) copied,
8.61276 s, 122 MB/s
```

что вполне типично для старого ржавого вращающегося диска (скорость тоже варьируется в зависимости от «блина»). Теперь можно обработать этот файл, чтобы измерить скорость чтения диска, но следует быть осторожными: файл или его части могут кэшироваться, так что вначале мы велим ядру очистить буферы перед выполнением теста на чтение:

```
$ echo 3 | sudo tee /proc/sys/vm/drop_caches
$ dd if=testfile of=/dev/null bs=1M count=1000
```

Перезапустите вторую команду, чтобы увидеть, почему необходима первая — наша система сообщила о замечательном (и ложном) результате в 7 ГБ/с, не очистив вначале кэш.

Результаты, полученные от *dd*, бывают чуть медленнее истинных возможностей устройства, поскольку ему могут мешать файловые системы и фрагментация. Чтение прямо с устройства устраняет это замедление, и утилита *Gnome Disks* позволяет сделать это. *Disks* включена в *Ubuntu* и большинство дистрибутивов с *Gnome*. Пакет обычно называется *gnome-disk-utility* — на случай, если вам придется установить его вручную. Запустите ►

Результаты, полученные от dd, бывают чуть медленнее истинных возможностей устройства.

зуемого для указания альтернативной системы *init*. Однако в этом случае обычный процесс *init* (т. е. */sbin/init*) отвечает, тогда как *bootchart* подсчитывает, изучает и измеряет множество переменных. Итак, чтобы вызвать при следующей загрузке *bootchart*, отредактируйте */boot/grub/grub.cfg*, добавив *init=/usr/lib/systemd/systemd-bootchart* к строке, загружающей ваше ядро (она начинается с *linux*). Если всё идет хорошо, то загрузка завершится, и в директории */run/log/* должно появиться SVG-изображение, в файле *bootchart-\$DATE.svg*. Обычно оно не очень хорошо прорисовано (метки

си. Вначале смонтируйте диск и через *cd* войдите на нем в директорию, для которой у вас есть доступ на чтение и запись (или станьте *root*, если уверены в своей компетенции). Мы собираемся записать 1024-МБ файл, полный нулей, неиссякаемый запас которых есть в устройстве */dev/zero*:

```
$ dd if=/dev/zero of=testfile bs=1M count=1000 conv=fdatasync
```

Обратите внимание, что опция *fdatasync* предоставляется через параметр *conv*. Она удостоверяет, что данные действительно записаны, а не остались в буфере до завершения процесса. Про нашу

Вы знаете, что действительно шлифует мою glxgears?

Утилита *glxgears* часто упоминается в связи с тестированием функциональности *OpenGL*. Если вы никогда не встречались с ней ранее (да даже если и встречались), она делает рендеринг трех шестеренок, за вращением которых очень приятно наблюдать. Она также выдает данные о FPS [частота кадров] в *stdout*, но поскольку большая часть настроек приковывает частоту кадров к частоте обновления монитора (так называемый *vsync*), обычно так отображается нечто очень близкое к 60 fps. В наше время для поддержания подобной частоты обновления не требуется много мощности GPU (или

даже CPU, кстати говоря), поэтому ценность *glxgears* для сравнительного тестирования сомнительна, если только вы не питаете слабость к шестеренкам. Но если вы используете драйверы, созданные в среде *Gallium* (т. е. *Nouveau* или драйверы с открытым кодом AMD, а не проприетарные драйверы или *Intel*), вы можете использовать *glxgears* для теста малоизвестной функции *Gallium Head Up Display* (HUD).

```
GALLIUM_HUD="fps.cpu+cpu0+cpu1:100; VRAM-usage;ps-invocations.primitivesgenerated" Glxgears
```

Получится прекрасное наложение в стиле *Fraps* — *Steam Settings* также предлагает базовый

счетчик *In-Game > FPS* — показывающий FPS, использование CPU и VRAM помимо всего прочего. Его можно индивидуально настраивать, и он работает с любой программой *OpenGL*, включая *Steam*. (Запустите *Steam* из командной строки, настроив там переменную *GALLIUM_HUD*. Он изобразит все, что обычно выдает клиент *Steam*, но вы не увидите его при запуске игры, и это будет намного проще, чем разбираться, как запускать игру из командной строки.)

Для более подробной информации установите переменную среды на "help".

программу и выберите диск, который хотите проверить, затем нажмите на две маленьких шестеренки (опции дополнительного раздела) под диаграммой Volumes. Выберите Benchmark Partition, а затем Start Benchmark.

В этот момент вы увидите, что доступен тест на запись, но тест требует, чтобы вы размонтировали устройство, и не будет работать в разделе, содержащем вашу ОС. Более того, его не стоит использовать на устройстве, содержащем данные без резервных копий — тест записывает на всем устройстве случайные данные, и хотя в теории потом он возвращает исходные данные, отключение питания или сбоя системы может помешать этому, так что не попадитесь в эту ловушку. Неразрушающий тест на чтение провести легко — предоставленные в примере числа и размеры по умолчанию вполне подойдут, просто нажмите кнопку Start. Диаграмма и графики разброса будут обновляться по мере проведения теста, показывая время доступа и скорость чтения соответственно.

Пока что для проведения сравнительного теста мы использовали инструменты базового уровня; для сравнения, у пользователей Windows имеются самые разные инструменты, пакеты, демо и прочие полнофункциональные инструменты для

измерения скорости своей системы. *Cinebench*, *CrystalDiskMark*, *Catzilla*, *FurMark* и пакет Futuremark (*3DMark*, *PCMark* и *VRMark*) часто используются для проверки самого передового оборудования на классных технических сайтах, но, увы, у них у всех нет доступных версий Linux. Некоторые из них можно заставить работать в *Wine*, однако вряд ли это позволит вам получить хотя бы подобие

прекрасных сцен и предлагает прелестные мини-игры. Другие популярные сравнительные тесты *Unigine*, *Valley* и *Heaven*, можно бесплатно скачать с <https://unigine.com/en/products/benchmarks/>. Расширенная и профессиональная версии доступны за определенную плату (во втором случае весьма солидную), и предлагают дополнительные функции, такие как создание отчета и коммерческое

Вы перейдете в магическое, красиво отрендеренное, стимпанковское царство.

верных измерений. *Cinebench* и *3DMark* в особенности ориентированы на DirectX, и если вы хотите получить обоснованные данные, придется повозиться с CSMT и/или с заплатками Gallium Nine для *Wine* и ваших графических драйверов. Это не входит в цели данной статьи. К счастью, есть очень милые (и бесплатные) программы, доступные для Linux.

Возможно, самая потрясающая из них та, которая вышла как раз когда мы это печатали: *Unigine Superposition*. Она использует движок Unigine 2 для проверки возможностей VP, делает рендеринг

применение. Несмотря на сложность сцен, рендеринг которых они проводят, заставить *Superposition*, *Heaven* и *Valley* работать легко. Просто скачайте файл `.run` с сайта, извлеките и запустите его. Например, чтобы запустить *Heaven*, сделайте следующее:

```
$ sh Unigine_Heaven-4.0.run
$ cd Unigine_Heaven-4.0
$ ./heaven
```

Появится меню, дающее доступ к ряду настроек. Весьма велик соблазн сразу перейти к качеству Ultra и сглаживанию x8, однако это способно спалить вашу видеокарту, так что для начала выберите что-нибудь помягче, а затем нажмите кнопку Run. Вы перейдете в магическое, красиво отрендеренное, стимпанковское царство, и частота кадров [fps] отобразится в верхнем правом углу. Каркасное моделирование и мозаичность можно переключать с помощью F2 и F3, а нажатие Esc откроет меню. Чтобы начать сам сравнительный тест, измеряющий минимальные и максимальные fps во всех сценах, нажмите F9. Как только он закончит, вы можете сохранить HTML-отчет о работе.

Phoronix Test Suite

Однако вершиной сравнительных тестов Linux является *Phoronix Test Suite* (PTS) Майкла Ларабеля [Michael Larabel]. Он расширяемый, автоматический, воссоздаваемый и с открытым кодом. Если вы хотите протестировать то, чего он еще не тестирует, можете написать собственные тесты в XML



► Наш компьютер не особо хорошо справился с раем *Unigine* — 5 fps на низком качестве без мозаики. Весьма посредственно.

Старина LINPACK

Один из самых старых сравнительных тестов CPU — это почтенный LINPACK. Изначально он был написан в 1970-х и являлся набором программ Fortran для линейной алгебры — операций с векторами и матрицами. Он предназначался для тестирования суперкомпьютеров той эпохи, а сейчас в основном его заменил LAPACK, который лучше использует векторные операции и прочие магические действия на современной архитектуре. Тем не менее, LINPACK остается актуальным, поскольку большая часть времени суперкомпьютеров сегодня по-прежнему тратится на инвертирование матриц. По сути, это не стандарт, поскольку существует несколько версий, которые можно найти на www.netlib.org/benchmark. Netlib — это репозиторий

работ и программ по математике, и на них определенно стоит обратить внимание, если вы занимаетесь научным программированием. Самая современная реализация LINPACK называется hpl (High Performance LINPACK) в Netlib. LINPACK по-прежнему используется для списка TOP-500 суперкомпьютеров, но ему требуются дополнительные библиотеки поддержки.

Вот как можно скомпилировать простой основанный на C LINPACK и измерить, сколько Операторов с Плавающей Запятой В Секунду [Floating point Operations per Second] (FLOPS) может обработать ваш компьютер.

```
$ wget http://www.netlib.org/benchmark/linpackc.new
-O linpack.c
```

```
$ gcc linpack.c -o linpack -lm
```

Используя массив по умолчанию 200×200, наш скромный компьютер обработал около 700 Мфлоп. Однако *gcc* — умное создание, и использование некоторых дополнительных оптимизаций компилятора, например,

```
$ gcc linpack.c -o linpack -lm -O3 -march=native
```

и перезапуск сравнительного теста изменил этот результат на 4,5 Гфлоп.

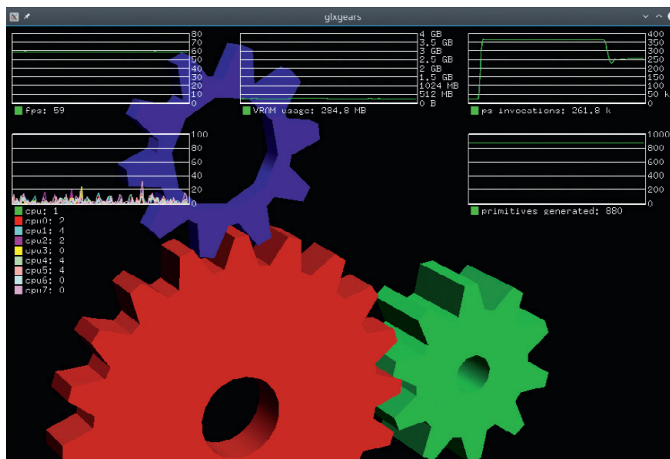
В *gcc* можно передать множество других оптимизаций, например, `-ffast-math` (что в определенных ситуациях также может быть `wrong-math`), и пользователям Gentoo они нравятся настолько, что они проводят долгие часы, компилируя с их помощью каждый пакет в своей системе.

и поделиться ими с сообществом. Результаты можно загрузить на сайт openbenchmarking.org, так что вы можете сравнить свои результаты с другими системами в прекрасно отрендеренных графиках. Установить PTS на системы на Debian легко: просто скачайте файл **.deb** с <http://phoronix-testsuite.com>. Его можно установить с помощью *Gdebi* или старого доброго `dpkg -i phoronix-testsuite_7.0.1_all.deb`. Наличествует ряд PHP-зависимостей, и если `dpkg` жалуется, их можно установить с помощью `apt install -f`. Есть ряд предварительно упакованных в пакеты тестов, способных проверить на скорость практически всё, что вы можете вообразить, и сделать это весьма надежно и рационально. Сюда входят классические программы Linux, например, для компиляции ядра (которые зависят от ввода/вывода диска и мощности процессора), синтетические сравнительные тесты, например, семейство *Unigine*, упомянутое выше, и игры — как с открытым кодом, так и доступные на Steam.

Предположим, вы хотите выполнить вышеупомянутый сравнительный тест компиляции ядра. Это сводится всего лишь к

```
$ phoronix-test-suite benchmark build-linuxkernel
```

PTS скачает исходники ядра (на данный момент — 4.9), установит все необходимые зависимости (*gcc*, *make* и т. д., он умеет делать это для разных дистрибутивов) и спросит вас, хотите ли вы загрузить ваши результаты. Затем он старательно скомпилирует ядро три раза (или даже больше, если обнаружится значительное несоответствие между компиляциями) и сообщит о среднем. У скромного компьютера LXF ушло около пяти минут; замечательный компьютер Ryzen, с которым мы баловались в LXF223, справился с этим всего за 78 с. Есть множество доступных тестов (просто запустите `phoronix-testsuite list-available-tests`), но некоторые из них не были обновлены или вообще не работают. Для сравнительного теста CPU хорошо подходит *fftw* (the Fastest Fourier Transform in the West), жизненно необходимый для цифровой обработки



► *Glxgears* уже не столь полезен в качестве сравнительного теста, но хорош для тестирования Gallium HUD с соответствующим оборудованием и драйверами.

сигналов. *John the Ripper* (взлом паролей), *encode-flac* (кодирование аудио) и *c-ray* (трассировка лучей) тоже неплохи. Для измерения ввода/вывода диска мы рекомендуем *IOzone* или *Dbench*. PTS поддерживает тестирование ряда игр, как открытых (например, *Xonotic*, *SuperTuxKart* и *Open Arena*), так и проприетарных из вашей библиотеки Steam (*BioShock Infinite*, *Mad Max*, *Civilization VI* и многие другие). Похоже, что PTS путается в любых основанных на Steam сравнительных тестах, если в фоне не запущен *Steam*, так что вначале запустите его, а затем — PTS из командной строки.

Воспроизводимость сравнительных тестов PTS делает их полезными для сравнения вашей системы с другими. Например, чтобы сравнить вашу чудесную установку с дряхлым и пропыленным устройством автора по трассировке лучей, кодированию аудио, открытому ключу, квадратному PHP-bench, просто запустите

```
$ phoronix-test-suite benchmark  
1704127-RI-LXFBENCHM51
```

Или, чтобы просто посмотреть результаты, загляните на сайт <http://openbenchmarking.org/result/1704127-RILXFBENCHM51>. Когда мы добавили ре-

зультаты PHPBench, что-то пошло криво, и PTS решил, что мы использовали другую систему. Это было отнюдь не так, поэтому сравните свои результаты с нашими и обязательно напишите нашему начальству: пусть убедится, что нам нужны более быстрые компьютеры. Много.

Верите или нет, но мы действительно лишь бегло коснулись того, что можно подвергнуть сравнительному тесту и как это можно сделать надежным способом. Стоило бы затронуть создание собственных тестов и пакетов в PTS, но у нас для этого просто нет места. Также не помешает взглянуть на такие программы, как *hardinfo*.

Для базового нагрузочного испытания запуска сравнительных тестов, нагружающих CPU в течение расширенного периода времени, будет вполне достаточно. Поиск Great Internet Mersenne Prime тоже не становится хуже (превращая его в долгие всего работающую программу распределенных вычислений в истории), а связанная с ним программа Prime95 (доступная под свободной лицензией) имеет режим неучастия [non-participation mode], идеально подходящий для тестирования выносливости вашей системы. LXF



► Gallium HUD можно заставить работать и со Steam... да, мы тоже не знаем, что такое «вызовы ps».

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!

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Перчинка и Морковка

Достойна только та жизнь, которая
прожита ради других людей.
Альберт Эйнштейн

Французский художник Давид Ревуа [David Revoy] считает, что индустрия комиксов должна быть серьезно отформатирована под современные реалии. Он создает свой web-комикс *Paper&Carrot* (милые истории, пригодные для всех, о начинающей ведьме Перчинке и ее коте Морковке) исключительно с помощью свободных инструментов *Krita* и *Inkscape*. Финансирование комикса происходит через краудфандинговую систему Patreon, где фанатам предоставляется возможность гарантировать художнику разумный ежемесячный доход.

В ответ сообществу представляется не только сам комикс, но и его исходники в формате Krita Document под лицензией Creative Commons Attribution 4.0. Благодаря этому web-комикс переведен как минимум на 28 языков совершенно официально, а Николай Мамашев из Горно-Алтайска собрал примерно \$4 тысячи на краудфандинговом сервисе IndieGogo под создание анимированной версии этих магических историй. К моменту публикации колонки анимированный комикс будет уже доступен широкой общественности. У меня чуть ранее уточнили, как мое имя должно выглядеть в титрах. Приятно, что есть такие люди, которым приятно помогать.

PS Теперь уже совсем официально: *Skype 4.3* для GNU/Linux прекратит работу с 1 июля 2017 г. E.m.Baldin@inp.nsk.su



В этом месяце вы научитесь...

- Изучать историю** 46 **Менять профили** 48
Bash старательно записывает все действия **Джейсона Кэннона** в терминале, чтобы при необходимости попользоваться ими снова — не пропадать же добру.
Ник Пирс обращается к *Gnome-Terminal*, чтобы придать своим разнообразным задачам узнаваемую индивидуальную раскраску и более в них не путаться.
- Настраивать роутер** 50 **Рисовать графы** 52
 Да, роутер и брандмауэр в самом деле получается настроить без особенной мороки, утверждает **Афнан Рехман**. И тут же доказывает это на практике.
 Представление в виде графа поможет визуализировать различные структуры, в том числе хэш-таблиц. **Михалис Цукалос** создает графы с помощью пакета Graphviz.
- Балансировать нагрузку** 56 **Доверять кому надо** 60
 Применяя обратный прокси-сервер, **Михалис Цукалос** перераспределяет задачи между компьютерами, чтобы обеспечить высокую доступность.
 Прежде чем использовать ключ шифрования, надо убедиться, что его создатель действительно является тем, кем себя называет. **Джон Лэйн** ведет расследование...
- Направлять трафик** 64 **Шифровать файлы** 68
 Не подумайте, что **Афнан Рехман** избегает трудностей: теперь он берется превратить в роутер дистрибутив Linux. Сохраним полный контроль над сетью!
Нейту Дрейку это удастся всего за три шага: файлы упаковываются в контейнер, защищенный паролем, а их оригиналы безопасно и бесследно удаляются.
- Применять AJAX** 70 **Воевать с роботами** 74
 Человечество не зря изобретало AJAX: **Максим Черепанов** употребил его для управления своим Послушным Домом.
 А зачем, собственно, с ними воевать? Пусть лучше сами роботы передерутся друг с другом, а **Лес Паундер** будет наблюдать за драчкой со стороны.

АКАДЕМИЯ КОДИНГА



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

Эй, Django! 78

Томас Рамболд разбирается в основах популярной среды разработки сайтов, а **Дэниел Сэмюэлс** ассистирует ему в написании кода.

Арсенал web-мастера 82

В Linux легко и бесплатно создаются и тестируются сайты и web-приложения! Чем активно пользуется **Кент Ельчук**.

Едва e4 86

Давненько не брали мы в руки шахмат... **Джонни Бидвелл** — тоже давненько, поэтому он решил сперва попрактиковаться на компьютерном движке.

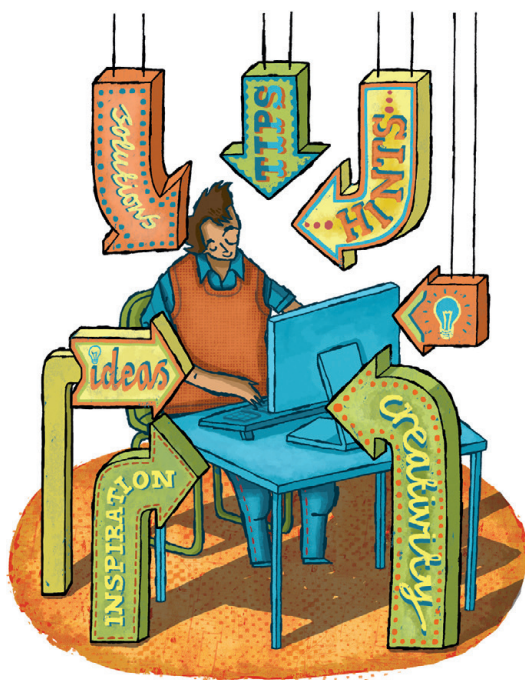
Terminal: Про diff и предысторию

Вместе с Джейсоном Кэнноном из UdeMy.com и курсов «Linux за пять дней» разберемся, как сравнивать файлы и пользоваться историей Bash.



Наш эксперт

Джейсон Кэннон начал карьеру как системный инженер Unix и Linux в 1999 г. С тех пор его навыки пригодились Xerox, UPS, Hewlett-Packard и Amazon.



На этом уроке мы узнаем, как сравнивать содержимое файлов. Если надо сравнить два файла и отобразить различия, можно воспользоваться командой *diff*. А также *sdiff* или *vimdiff*. Команда *diff* позволит увидеть, чем отличаются два файла, а команда *sdiff* отобразит содержимое первого справа, а второго — слева. *Vimdiff* отобразит различия с помощью редактора *vim*.

» Извлеките максимум из своей командной строки, используя немного истории.

```
lxf@lxf-HP-ProDesk-490-G2-MT: ~
31 kstart
32 rstart
33 exit
34 sudo shutdown -r 0
35 sudo apt-get install ubuntu-desktop
36 sudo apt-get autoremove
37 cp /etc/X11/xorg.conf /etc/X11/xorg.conf.bak
38 sudo rm /etc/X11/xorg.conf
39 echo "nouveau" | sudo tee -a /etc/modules
40 sudo reboot
41 sudo apt-get remove --purge nvidia*
42 sudo nvidia-settings
43 sudo apt-get remove --purge xserver-xorg-video-nouveau xserver-xorg-video-nv
44 sudo apt-get install nvidia-common
45 sudo apt-get install xserver-xorg-video-nouveau
46 sudo apt-get install --reinstall libgl1-mesa-glx libgl1-mesa-dri xserver-xorg-core
47 sudo apt-get install --reinstall libgl1-mesa-glx libgl1-mesa-dri xserver-xorg-core
48 sudo dpkg-reconfigure xserver-xorg
49 sudo shutdown -r 0
50 history
lxf@lxf-HP-ProDesk-490-G2-MT:~$
```

```
diff file1 file2 # сравниваем два файла
sdiff file1 file2 # файлы бок о бок
vimdiff file1 file2 # внутри vim
А вот первая строка вывода diff.
$ diff file1 file2
3c3
< this is a line in a file [это строка в файле].
---
> This is a Line in a File [Это Строка в Файле]!
```

Первое число в `3c3` представляет номер строки из первого файла, а второе — номер строки из второго файла. Средним символом, разделяющим номера строк, будет либо `<` для изменения, либо `|` для удаления, либо `>` для добавления. Таким образом, формат будет выглядеть так:

```
<line number in file1>-<action>-<line number in file2>
```

В приведенном примере файлы различаются, начиная с третьей строки. Вывод, следующий за знаком «меньше», относится к первому файлу. Текст, следующий за значком «больше», относится ко второму файлу. Три дефиса — это просто разделители.

В выводе *sdiff* символ в виде вертикальной черты означает, что текст файлов отличается в этой строке. Также там может быть знак «меньше», означающий, что строка присутствует только в первом файле. Знак «больше» означает, что строка присутствует только во втором файле.

```
$ sdiff file1 file2
line in file1 | line in file2
> more in file2
| различающиеся строки
< строка есть только в file1
> строка есть только в file2
```

При запуске *vimdiff* файлы будут отображаться каждый в своем окне. Вы можете использовать стандартные элементы управления *Vim* [Ред.: — То есть бред сумасшедшего из прошлого столетия], такие как `:q` для выхода из текущего окна, `:qa` для выхода из всего и `:qa!` для принудительного завершения. Для переключения окон воспользуйтесь `Ctrl+W`.

Принцип работы этих команд мы можем увидеть на примерах, приведенных ниже. Воспользуемся *cat*, чтобы отобразить их с номерами строк:

```
$ cat -n secret
1 tags: credentials
2 site: facebook.com
3 user: jason
4 pass: Abee!
5 tags: credentials
$ cat -n secret.bak
1 tags: credentials
2 site: facebook.com
```


Учитесь и экономьте с Udemу

Если вам понравилась эта маленькая дегустация курса «Linux за пять дней» от Udemу, неограниченный доступ к полному курсу вы можете получить на udemy.com с эксклюзивной скидкой для читателей *Linux Format*. Вы узнаете всё об использовании стандартных инструментов терминала для администрирования, создании тестовой машины, о правах доступа к файлам Linux, стандартных текстовых редакторах, манипулировании файлами, использовании сетевых передач, управлении процессами и многом другом.

Получите скидку, начните учиться!

- » Посетите сайт www.bit.ly/LearnLinux5, чтобы записаться на курс по сниженной цене £ 15 (скидка 92%, действительна до 6.06.2017)! Нажмите кнопку Buy Now [Купить сейчас] и зарегистрируйтесь для получения учетной записи на Udemу.
- » Когда вы зарегистрируетесь, вам предложат подтвердить вашу покупку.
- » Курс со скидкой в размере £ 15 будет выбран автоматически при использовании указанной выше

ссылки. Введите данные своей кредитной карты и нажмите Pay Now [Оплатить сейчас].

» Вы успешно поступили на курс! Можете наслаждаться неограниченным доступом.

Udemу была основана в 2010 г., чтобы улучшать жизнь посредством обучения.

Udemу — это глобальный рынок обучения и преподавания в Интернете, где более 15 миллионов студентов обучаются на библиотеке из 45000 курсов, преподаваемых экспертами-инструкторами на 80 различных языках.

```
3 user: jason
4 pass: bee
5 tags: credentials
$ diff secret secret.bak
4c4
< pass: Abee!
---
> pass: bee!
```

Вы заметите, что строка, начинающаяся с символа «меньше», относится к первому файлу, а строка с символом «больше» относится ко второму файлу. Вы также заметите, что первая строка вывода *diff* выглядит как `4c4`. Это означает, что четвертая строка первого файла изменилась или отличается от четвертой строки второго файла.

```
$ echo new last line >> secret
$ sdiff secret secret.bak
tags: credentials          tags: credentials
site: facebook.com        site: facebook.com
user: Jason                user: jason
pass: Abee!                | pass: bee
tags: credentials         tags: credentials
new last line              <
```

Здесь вы можете увидеть *sdiff* в действии. Он размещает файлы друг напротив друга, а символ вертикальной черты указывает на строку, где имеются различия. Мы добавили в файл новую последнюю строку. Символ «меньше» показывает, что в первом файле есть строка, которой нет во втором.

История оболочки

Давайте рассмотрим историю команд оболочки, повторно выполняя команды или части команд с помощью символа восклицательного знака. Каждая команда, которую вы вводите в оболочку, регистрируется в истории оболочки, и иметь доступ к ней чрезвычайно полезно. Вы можете выполнять поиск по истории, повторять ранее введенные команды, а также вызывать их, а затем изменять перед выполнением.

Это поможет не только сэкономить время и нажатия клавиш, но и избежать ошибок за счет повторения уже известных правильных команд. Некоторые оболочки, такие как *Bash*, хранят историю в ОЗУ и записывают ее в файл только при выходе.

Общие файлы истории — это `.bash_history`, `.history` и `.histfile`. Эти файлы истории хранятся в вашем домашнем каталоге. Команда *history* отображает команды в истории оболочки, перед каждой из них должен быть номер, по которому ее можно будет впоследствии найти. По умолчанию *Bash* хранит в истории 500 команд. Это число контролируется переменной среды `HISTSIZE`.

```
!N # повторить команду N
!! # повторить последнюю команду
! # повторить самую недавнюю команду <string>
```

Синтаксис *history* можно расширить, добавив символ восклицательного знака для повторного выполнения команды по номеру. Для получения списка команд с их номерами запустите команду *history*. Желая повторно запустить команду номер три из своей истории, вы должны ввести `!3`, а затем нажать Enter. Если надо повторить строку, начинающуюся с определенной команды или даже с символа, можно использовать комбинацию `!`. Например, для повтора команды `cat`, которую вы недавно выполняли, вы можете просто набрать `!c` и нажать Enter.

Помимо запуска целых команд в своей истории, вы можете вызывать отдельные части командной строки. Синтаксис выглядит как `!:<NUM>`, где `!` обозначает событие, т.е. последнюю команду. Можно использовать восклицательный знак, о котором мы только что говорили, т.е. `!<num>!!` или `<string>`. Параметр `:<NUM>` обозначает слово в командной строке, 0 — это запуск команды, 1 — первый аргумент, и т.д.

```
$ head file.txt sort.txt note.txt
<вывод из файла>
$ !!
head file.txt sort.txt note.txt
<вывод из файла>
$ vi !:2
vi sort.txt
```

В примере выше `!:0` — заголовок, `!:1` — `file.txt`, `!:2` — `sort.txt`, а `!:3` будет `note.txt`.

Есть еще два полезных ярлыка `!`, которые следует знать:

```
!^ # первый аргумент
!$ # последний аргумент
$ head file.txt sort.txt note.txt
!^ = file.txt
!$ = note.txt
```

Это всего лишь малая толика того, что есть в курсе «Linux за пять дней от Udemу». Чтобы узнать больше, обратите внимание на специальное предложение для читателей *вверху!* **LXF**

```
secret          1,1          ALL secret.bak          1,1
```

Скорая помощь

Восклицательный знак (!) иногда называют bang.

Скорая помощь

Чтобы использовать *vimdiff*, нужно сначала установить *Vim*, с помощью `sudo apt-get install vim`.

» Установив *Vim*, вы сможете использовать для сравнения файлов *vimdiff*.

Terminal. Когда профилей много

Ник Пирс показывает, как настроить терминал для различных целей с помощью пользовательских профилей.



Наш эксперт

Ник Пирс уже более 30 лет балует с компьютерами, из них почти десятилетие — с Linux.

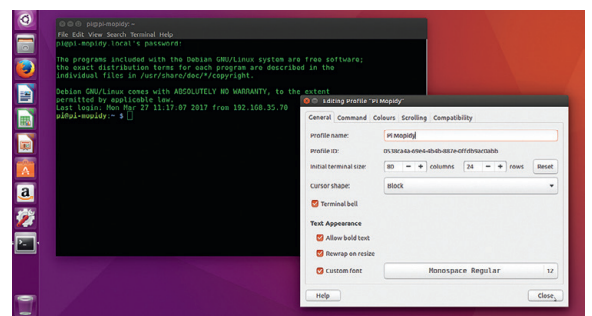
По умолчанию терминалом в Ubuntu является *Gnome-Terminal*, а одна из его малоизвестных функций — поддержка профилей. Профили представляют собой набор параметров терминала, включая имя профиля, цвета шрифта и фона и прокрутку. Кроме того, профили можно настроить на запуск при старте определенной команды или оболочки, открытие на конкретном экране и даже подключение к удаленному компьютеру через SSH. Эта гибкость позволяет легко понять, зачем одному пользователю может потребоваться создать более одного профиля для использования терминала — вы, например, можете регулярно администрировать через SSH другой ПК, скажем, Pi Zero без графического интерфейса с работающим на нем музыкальным сервером *Mopidy* (см. [LXF218](#)). В этом случае можно создать профиль, который сразу подключает вас к Pi Zero и использует другую цветовую схему, чтобы помочь вам отличить это окно терминала от других, которые вы также можете запускать.

Создайте и управляйте профилями

В нарушение правил, вы не можете использовать командную строку для создания и управления профилями — инструмент *dconf* можно использовать из оболочки после настройки профиля, но только для просмотра и изменения настроек, установленных по приведенным ниже советам и рекомендациям. Во врезке показано, что можно сделать с помощью инструмента *dconf*.

Итак, чтобы создать свой первый пользовательский профиль, надо открыть терминал, затем выбрать **File > New Profile** [Файл > Создать профиль]. Появится диалоговое окно, состоящее из пяти вкладок. Начните с выбора вашему профилю подходящего описательного имени, которое вам впоследствии пригодится для выбора при переключении профилей.

Прежде чем устремляться дальше, нажмите кнопку **Close** [Заккрыть], затем переключитесь на новый профиль через меню



» Профили создаются и настраиваются с помощью диалогового окна **Profile Preferences** [Параметры профиля], доступного из меню **Edit терминала**.

Terminal > Change Profile [Терминал > Изменить профиль]. Теперь выберите **Edit > Profile Preferences** [Правка > Параметры профиля], чтобы вновь открыть диалоговое окно **Editing Profile** [Изменение профиля] — вы войдете с новым профилем с учетом изменений, поскольку любые ваши поправки применяются в реальном времени и сразу появляются в окне терминала. Эксперименты с цветовыми схемами, разными шрифтами и т. д. для пытливых обязательны!

Вкладка **General** также содержит параметры начального размера окна терминала в столбцах и строках — по умолчанию — 80×24. Можно также изменить форму курсора с основного — **Block** [кирпичик] на **I-Beam** [I-образный] (прямая мигающая вертикальная линия, как в *LibreOffice Writer*) или **Underline** [Подчеркивание] — форма курсора меняется сразу при выборе.

Снимите флажок **Terminal Bell** [Подать гудок], если не хотите получать звуковые уведомления через внутренний динамик ПК; к сожалению, на визуальные (т. е. мигающие) уведомления для определенных профилей переключиться нельзя. Последние три параметра на вкладке **General** [Общие] самоочевидны и начинаются с установки запрета жирного шрифта. Также можно указать, разрешать ли тексту автоматически перераспределяться при ручном изменении размера окна терминала, а также выбрать собственный шрифт для отображения.

Инструмент *dconf* показывает, что параметр пользовательского шрифта содержит две отдельные настройки. Первый — **“use-system-font”** — определяет, используете ли вы системный шрифт или нет, всего лишь значениями **true** (не отмечено) или **false** (отмечено). Второй — **“font”** — работает только когда **“use-system-font”** установлен в **false**, и определяет заменяемый шрифт, стиль и размер.

Другие параметры настройки

Переключитесь на следующую вкладку **Colours** [Цвета], чтобы задать цвета текста и фона — снимите флажок **Use Colours From System Theme** [Использовать цвета из системной темы], и сможете

Скорая помощь

Пользуясь *dconf*, установили неподдерживаемое значение ключа? Настройки по умолчанию восстанавливаются так: `$ dconf reset /org/gnome/terminal/legacy/profiles:/<profileID>/<keyname>`.

Переключение профилей

Профили легко менять, даже не выходя из текущего сеанса работы с терминалом: очевидный выбор — открыть меню **Terminal > Change Profile** и выбрать нужный профиль. При этом настройки данного профиля сразу применяются к текущему сеансу. Однако никакие пользовательские команды, назначенные этому профилю, не запустятся.

Можно также запустить новый сеанс **Terminal** из командной строки — обратите

внимание, он откроется в новом окне (соответствует выбору **File > Open Terminal** [Файл > Открыть терминал]):

```
$ gnome-terminal --window-with-profile=<PROFILENAME>
```

Замените **<PROFILENAME>** выбранным вами именем профиля (с учетом регистра. Если в имени профиля есть пробелы, поместите перед пробелом символ обратной косой черты — например, `Mopidy\ server`).

Использование командной строки

Профили терминала управляются инструментом `dconf`; это инструмент низкого уровня для управления конфигурациями и системными настройками. Вы можете применять его для чтения значений любых параметров (называемых ключами), которые вы преднастроили через диалоговое окно Profile Preferences. Сначала запишите Profile ID выбранного профиля на вкладке General в диалоговом окне Profile Preferences [Изменение профиля]. После

идентификации вернитесь в командную строку и введите `$ dconf list /org/gnome/terminal/legacy/profiles:<profileID>`, подставив идентификатор своего профиля (включая символ двоеточия впереди) вместо `<profileID>`. Появятся только уже настроенные ключи. Просмотреть текущее значение конкретного ключа поможет команда `dconf read`:
`$ dconf read /org/gnome/terminal/legacy/profiles:<profileID>/exit-action`

Замените `exit-action` выбранным ключом.

`dconf write` послужит для изменения значения ключа, если вы знаете допустимые значения:

```
$ dconf write /org/gnome/terminal/legacy/profiles:<profileID>/exit-action "restart"
```

Используя `dconf`, новые ключи создавать нельзя. Некоторые из правильных значений — довольно простой выбор между 'true' и 'false', но есть и сложнее. При проблемах — см. «Скорая помощь».

быстро переключиться на другую встроенную схему (например, зеленый текст на черном фоне) или вручную задать цвета, используя различные цветовые палитры. Также можно включить прозрачность и применить ползунок для смешения окна терминала с фоном. В разделе Palette [Палитра] задаются индивидуальные цвета — опять же, можно выбрать между встроенными схемами или Custom [Другая], чтобы назначить все цвета по отдельности. Вкладка Scrolling [Прокрутка] управляет поведением терминала четырьмя разными способами: во-первых, можно выбрать отображение или скрытие полосы прокрутки (если ее скрыть, придется прокручивать исключительно колесиком мыши или жестом трекпада). Scroll on Output [Прокручивать при выводе] по умолчанию отключено, и можно прокручивать вверх с помощью мыши, чтобы остановить автоматическую прокрутку при большом объеме вывода. Включение этой опции отключает такую возможность. Опция Scroll on Keystroke [Прокручивать при нажатии клавиши] отмечена флажком и работает аналогично Scroll on Output, за исключением того, что она связана с фактическими нажатиями клавиш, и по этой причине лучше оставить ее как есть. Наконец, Limit Scrollback To [Обратная прокрутка]: показывает, сколько строк вывода терминала хранится в памяти — этот буфер очищается при каждом перезапуске сеанса терминала, и вы, возможно, захотите уменьшить это число, если у вас мало памяти, а вам надо создать большие объемы вывода без ссылки на предыдущие строки. Если вы решите не ограничивать его вообще — такое не рекомендуется, кроме случаев, когда у вас достаточно свободного ОЗУ — то установите цифру 0, но остерегайтесь потери производительности. Вкладка Compatibility [Совместимость] позволяет изменить поведение клавиш Backspace и Delete, а также установить кодировку по умолчанию для сеанса терминала. Последний параметр — Ambiguous-width Characters [Символы неоднозначной ширины]: — можно установить в Wide [Широкие], если вы работаете с определенными языками или символами вроде греческих или азиатских логограмм.

Команды пользователя

Принарядив свой профиль, пора перейти на вкладку Commands [Команды], где вы найдете два флажка. Чтобы запустить простую команду при запуске профиля, поставьте флажок Run a Custom Command Instead of My Shell [Запустить другую команду вместо моей оболочки], а затем введите свою команду в поле Custom Command [Другая команда]:. Так, для входа на удаленный ПК —

```
ssh user@compname
```

Выпадающее меню When Command Exits [При выходе из команды]: позволяет задать, что произойдет по завершении команды — по умолчанию окно терминала закроется, но можно также выбрать Hold the Terminal Open [Держать терминал открытым] или Restart the Command [Перезапустить команду]. Если вы запускаете простую команду типа `sudo apt-get update`, первое — разумный выбор.

Обратите внимание, что при использовании пользовательской команды для удаленного входа на другой компьютер действие

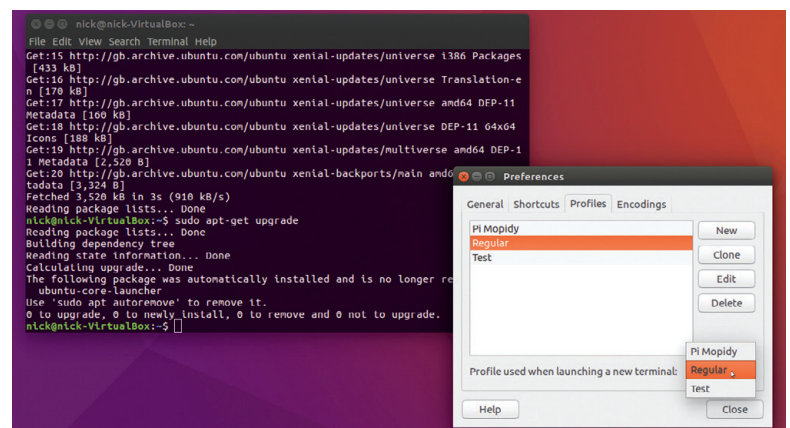
When Command Exits: выполняется до закрытия соединения вручную, обычно с помощью команды Exit. Выберите, например, Hold the Terminal Open, и вы увидите всплывающее меню с удобной кнопкой Relaunch [Перезапуск], если вы хотите по какой-либо причине немедленно перезапустить соединение. На вкладке Command еще имеется опция Run Command as a Login shell [Запускать команду как оболочку входа] — отметьте ее, чтобы терминал прочитал файл `.profile` — основной файл инициализации всей системы, выполняющийся при входе в непосредственно оболочку — а не файл `.bashrc`, используемый при открытии терминала через рабочий стол Unity. Вы поймете, если или когда вам это понадобится.

Завершим настройки

На этом заканчивается процесс создания и редактирования вашего первого профиля. Когда вы в следующий раз откроете терминал, он по умолчанию будет использовать ваш основной профиль, но вы сможете легко переключиться, следуя советам во второй врезке. Нужен еще один профиль? Выберите File > New Profile еще раз, чтобы создать его на основе текущего профиля, с тем же названием и настройками, готовыми к правке. Желая создать новый профиль с нуля, выберите Edit > Preferences [Правка > Параметры] и перейдите на вкладку Profile [Профили]. Здесь вы увидите список всех имеющихся профилей — нажмите New [Создать], чтобы создать новый пустой профиль, или выберите профиль из списка и нажмите Clone [Клонировать], чтобы создать его на основе этого шаблона. Кроме того, можно редактировать профили без необходимости сначала переключаться на них, а также удалять ненужные профили.

Последнее, но не по значению: здесь внизу также есть выпадающее меню — Profile Used When Launching a New Terminal [Профиль, используемый при запуске нового терминала]:. Используйте его, чтобы выбрать профиль по умолчанию при запуске терминала (или в новом окне терминала) в будущем. **LXF**

Через месяц:
Работа онлайн



➤ **Задайте свой профиль по умолчанию с помощью этого диалогового окна — здесь вы также можете управлять своим набором профилей.**



pfSense: Личный брандмауэр

Афнан Рехман показывает, как легко создать собственную систему роутера и брандмауэра с помощью этой программы с открытым кодом.



Наш эксперт

Афнан Рехман — студент, энтузиаст Linux и компьютерный умник, который с ходу всё ломает, так что вам этого делать не придется.



Скорая помощь
Если вы делаете установку на 64-битное оборудование, убедитесь, что взяли 64-битную версию pfSense. Установка 32-битной pfSense возможна, но она порой бывает нестабильна и не рекомендуется.

Вам никогда не хотелось создать свой собственный роутер без мороки с созданием собственных iptables и сетевых правил с самого нуля (как это проделывается на стр. 64)? Вполне возможно получить требуемые вам функции и производительность без всех этих титанических усилий. Решением является pfSense, дистрибутив сетевого брандмауэра/роутера с открытым кодом на основе ОС FreeBSD. pfSense используется во множестве разных приложений, от домашних роутеров до корпоративных решений, и к нему благожелательно относятся в сообществе благодаря его надежности и умениям. Если у вас есть для него оборудование, весьма скромное по любым стандартам, pfSense можно установить на любой компьютер и управлять из web-интерфейса на отдельном клиентском устройстве в той же сети. Всё, что вам нужно, это процессор получше Pentium II, 256 МБ или больше ОЗУ и как минимум 1 ГБ места на диске, а также не менее двух портов Ethernet: один для WAN и один для LAN.

Заготовив свое оборудование, перейдите на страницу скачивания pfSense на www.pfsense.org/download и выберите архитектуру. Если вы используете достаточно современный компьютер, лучше, вероятно, выбрать архитектуру AMD64. Нажмите на кнопку скачивания и выберите самую последнюю стабильную версию для скачивания. Как только у вас будет файл, мы советуем установить его через USB-привод, хотя CD тоже подойдет, если у вашего ПК роутера есть доступ к приводу оптических дисков. Во втором случае можно прожечь программу установки на CD, а в первом — записать образ на USB-устройство с помощью такого инструмента, как Rufus из Windows или команда dd для записи из Mac OS X или Linux.

По завершении записи вы готовы установить pfSense на ваш роутер. Подключите выбранное средство установки и загрузитесь с него, используя меню загрузки на вашем ПК роутера. Вы увидите текстовое окно приветствия, показывающее несколько опций настройки и таймер с обратным отсчетом.

Нажмите 'I' на клавиатуре, чтобы начать установку.

Когда вы окажетесь в программе установки, вам предложат несколько опций для задач, которые будут выполняться программой. Для большинства будет нормально выбрать опцию Quick/Easy Install [Быстрая/Простая Установка], которая предполагает, что первый диск является целевым диском для установки. Если на вашей системе роутера установлен только один жесткий диск, это не вызовет никаких проблем. Однако если у вас несколько устройств хранения, рекомендуем использовать опцию Custom Install [Индивидуальная установка], чтобы установить pfSense на подходящий диск. Далее программа установки спросит вас, принимаете ли вы изменения. Возможность отката отсутствует, и если на вашем целевом диске имеются некие данные, вы должны быть уверены, что их потеря не окажется проблемой. Как только вы нажмете OK, программа продолжит очистку диска и установку pfSense, что потребует некоторого времени. Можете прерваться на чашечку чая.

```
Configuring firewall.....done.
Generating BSD graphs...done.
Starting syslog...done.
Starting CRON...done.
pfSense (pfSense) 2.3.3-RELEASE i386 Thu Feb 16 06:59:50 CST 2017
Bootup complete

FreeBSD/i386 (pfSense.localdomain) (ttyv0)
*** Welcome to pfSense 2.3.3-RELEASE (i386 full-install) on pfSense ***
WAN (wan) -> em0 -> 4/ВНСП4: 10.0.2.15/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults 13) Update from console
5) Reboot system             14) Enable Secure Shell (ssh)
6) Halt system               15) Restore recent configuration
7) Ping host                 16) Restart PHP FPM
8) Shell

Enter an option: █
```

➤ После установки pfSense и перезагрузки компьютера, при условии, что вы удалили средство установки, программа загрузит меню, показывающее сетевые интерфейсы и опции для них.

Эксплуатация роутера

Будем надеяться, что к концу этого руководства у вас будет рабочий роутер с работающим pfSense и web-интерфейсом, доступ к которому легко получить через клиентский компьютер. У нас есть для вас несколько финальных предложений, когда вы достигнете этого момента. Во-первых, в конце у вас, скорее всего, не будет никакой необходимости в проверке дисплея прямо с компьютера, на котором работает pfSense. Если у него есть монитор, можете смело использовать этот монитор для иных

целей. Даже если вам понадобится перезапустить роутер, это можно сделать из web-интерфейса, как с обычным роутером, который вы контролируете через web-интерфейс. Поскольку у этого роутера нет возможностей беспроводного соединения, мы рекомендуем добавить нечто вроде точки беспроводного доступа к вашей сети для возможности подключения беспроводных устройств и переключателей для размещения нескольких проводных устройств. И то, и другое относительно дешево

и легко добавляется в систему. И, наконец, рекомендуем держать оборудование роутера в безопасном, надежном и удобном для доступа месте в доме на случай, если вам нужно будет до него добраться. Простые действия, например, периодическая протирка его от пыли, снизят шум и выделение тепла, что значительно продлит срок жизни вашего оборудования и позволит вам пожинать плоды работы высокопроизводительного и отлично настроиваемого роутера долгие годы.

Дополнительная помощь

В некоторый момент в процессе настройки вашей системы pfSense у вас могут возникнуть затруднения. Это нормально: всё же наш урок — не нудное исчерпывающее руководство по абсолютно всем аспектам pfSense. К счастью, есть множество онлайн-ресурсов, которые помогут вам по пути. pfSense — очень популярная и известная операционная система брандмауэра/роутера, и найти подробную документацию и подсказки по решению практически всех возможных проблем несложно. Одно из рекомендованных средств — официальная

документация pfSense на https://doc.pfsense.org/index.php/Main_Page. Документация от разработчиков достаточно полная и подробная, и в ней куча советов по решению большинства обычно встречающихся проблем. Охвачены установка, функции, версии релизов, ЧАВО, пакеты для расширения возможностей pfSense — и, конечно, особый раздел посвящен решению проблем. Кроме того, документация предлагает несколько руководств по включению и использованию расширенных функций, таких как удаленное администрирование брандмауэра.

Официальный сайт также предлагает советы по выбору оборудования и дальнейшему обучению, если вы решите взяться за более расширенные проекты, например, использование роутера pfSense в офисе, где доступ к сервисам роутера могут получать одновременно сотни людей.

Конечно, не стоит ограничиваться только решениями на официальном сайте: загляните также на форумы и поищите официальные руководства в Интернете, чтобы побольше узнать о своей проблеме и способах ее решения.

По завершении установки программа установки попросит вас выбрать между стандартным ядром и встроенным ядром. Если вы не очень уверены в своих действиях, рекомендуем выбирать стандартное ядро, разрешающее консоль VGA. Далее вам предложат перезагрузиться.

Перезагрузившись, вы опять увидите таймер загрузки и экран с интерфейсами, соответствующими портам Ethernet в вашей системе. Вспомните, ранее говорилось, что вам нужно не менее двух.

Вопросы, вопросы

Первый вопрос, который вам зададут — хотите ли вы настроить VLANs. VLANs — это Virtual Local Area Networks [Виртуальные Локальные Сети], и большая часть домашних пользователей ими не пользуется. Они могут быть полезны, если вы хотите разделить вещательные домены или изолировать трафик в целях безопасности. Обычно это используется в больших офисах и т.п. На данный момент, если только вам это не нужно, нажмите 'N', чтобы отказаться от настройки VLANs.

Два интерфейса по умолчанию — em0 и em1. Обычно нам нравится определять em0 как WAN (входящий трафик из Интернета), а em1 — как интерфейс LAN (исходящий из вашей локальной сети устройств). Введите это в соответствующие области, когда вас об этом попросят. Вам также предложат ввести имя опционального интерфейса, но это не обязательно, так что можете это пропустить, нажав на клавишу Enter. Затем программа попросит вас подтвердить настройки для интерфейсов LAN и WAN. Проверьте их правильность и затем нажмите Y, чтобы продолжить.

Затем операционная система определит интерфейсы и отобразит их IP-адреса вместе с некоторыми опциями и предложением выбрать опцию. IP-адрес WAN обычно присваивается через DHCP от вашего интернет-провайдера. IP-адрес LAN по умолчанию будет 192.168.1.1. Это можно изменить, выбрав опцию 2 для настройки IP-адреса каждого интерфейса. Если вы хотите использовать DHCP для клиентов на вашей локальной сети, можете также настроить приемлемый диапазон IP-адресов также с помощью опции 2. Выберите вторую опцию из главного меню, а затем выберите интерфейс LAN.

В опции, которая спросит о количестве бит новой LAN IPv4, введите число, соответствующее маске подсети, которую вы хотите применять, используя таблицу над окном. Для большинства домашних пользователей это будет 24. Затем, когда вас спросят, хотите ли вы включить DHCP-сервер на LAN, введите 'Y' и введите начальный и конечный адреса для диапазона адресов клиентов, которые хотите использовать. Обеспечьте, чтобы этот диапазон охватывал все устройства, которым нужны IP-адреса в вашей сети, и оставьте место на будущее.

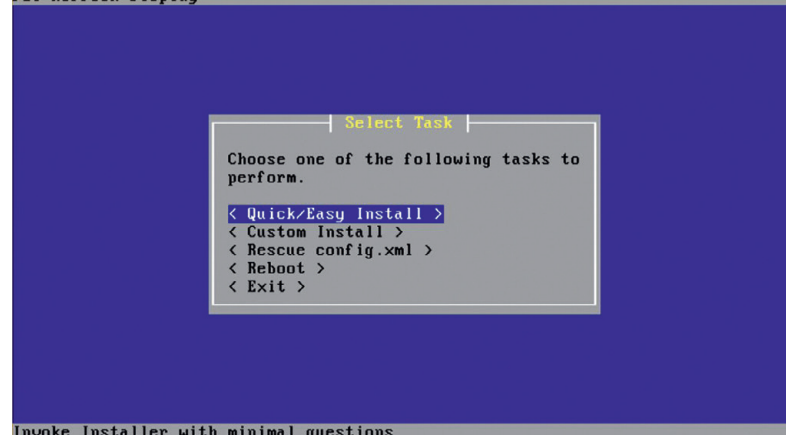
По завершении данной настройки вам выдадут ссылку на web-конфигуратор с клиентского устройства в той же сети. Используйте ее для входа на web-конфигуратор. С этого момента вам, скорее всего, не нужен будет доступ к консоли прямо с роутера: вы сможете получить доступ ко всему необходимому через web-конфигуратор.

На вашем клиентском компьютере web-адрес должен перевести вас на экран входа. По умолчанию имя пользователя и пароль — "admin" и "pfsense" соответственно. После первого входа в систему окно настройки проведет вас по процессу начальной конфигурации pfSense, включая ввод домена и имени хоста, DNS серверов и временного пояса и т.п. У вас также будет возможность настроить интерфейсы WAN и LAN, и после этого вам предложат изменить имя пользователя и пароль для администратора. Настоятельно рекомендуем воспользоваться этой возможностью и задать сильный пароль, дабы ваша сеть не была взломана.

Покончив с настройкой, система перезагрузится, применив ваши изменения. По завершении процесса перезагрузки вас ждет поздравление и доступ к панели управления web-конфигуратора pfSense. Из этого GUI вы можете добавлять расширенные настройки, например, фильтры MAC, настройку VPN и настройки брандмауэра, которые можете модифицировать на свое усмотрение.

Итак, вы настроили собственный роутер pfSense, который теперь можно использовать постоянно в вашей домашней сети. Можете спокойно перейти к более сложным вещам, например, добавлять в свою систему расширенные настройки или даже собрать с нуля собственный роутер. Возможности безграничны! **LXF**

F10=Refresh Display



» Во время начальной установки вы пройдете по относительно простому меню. Следуйте подсказкам, и установка должна пройти гладко.

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

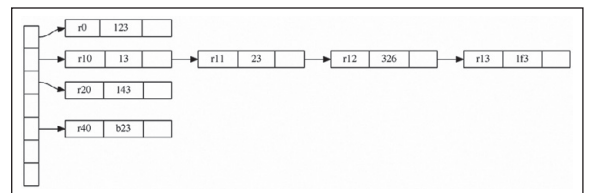
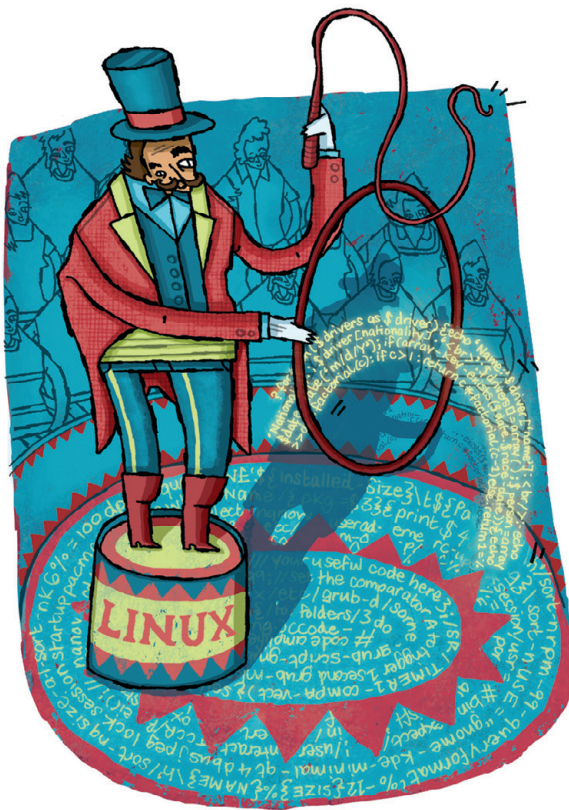
Graphviz: Граф будет красивым

Михалис Цукалос научит вас использовать Graphviz не только для создания красивых графов, но и программ, которые генерируют код Graphviz.



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.



» Вывод `hashtable.dot`, где можно видеть, как создаются хэш-таблицы с помощью языка `dot`.

Graphviz имеет собственный язык под названием DOT — простой, элегантный и эффективный. Graphviz хорош тем, что можно писать код в простом текстовом редакторе, и легко создавать скрипты, генерирующие код Graphviz!

На этом уроке будет показан простой код DOT, три программы и инструмент для преобразования кода DOT в LaTeX. Помимо представленных примеров, Graphviz можно также использовать для создания карт сайта, схем UML, схем связей между таблицами базы данных, потоков сетевых пакетов и структур данных любого типа. Помните: разобравшись, как работает язык Graphviz, вы сможете использовать для создания собственного кода Graphviz любой язык программирования, который вам нравится. Поэтому стоит начать изучать всё о Graphviz и о его возможностях.

Инструменты командной строки

В системе Debian Linux Graphviz можно установить следующим образом:

```
# apt-get install graphviz graphviz-dev graphviz-doc
```

Graphviz поставляется с множеством инструментов командной строки, таких как `dot`, `neato`, `twopi`, `circo`, `fdp`, `sfdp` и `patchwork`.

Инструмент `dot` используется для рисования ориентированных графов, тогда как `neato` отвечает за рисование неориентированных графов — это два наиболее часто используемых инструмента пакета Graphviz. Инструмент `twopi` также бывает очень удобен, так как он позволяет определять и легко изменять центральный элемент вашего графа, что достигается с помощью команды `center`, за которой следует имя узла. Не забудьте поставить команду `center` в начале своего файла DOT. Дополнительную информацию о других инструментах можно найти на tap-странице `dot`. Еще больше о Graphviz можно узнать, посетив www.graphviz.org.

Следующий код Graphviz, который мы сохранили как `simple.dot`, проверит, что GraphViz у вас установлен и работает отлично:

```
digraph G
{
graph [ dpi = 300 ];
```

Скорая помощь

Вы можете больше узнать о Graphviz, прочитав *Graph Drawing Software, Series: Mathematics and Visualization*, а о графах — прочитав отличную книгу *The Design and Analysis of Computer Algorithms*, написанную Ахо, Хопкрофтом и Ульманом [Aho, Hopcroft, Ullmann].

Graphviz — это набор инструментов для создания ориентированных или неориентированных графов. Согласно книге Альфреда В. Ахо [Alfred V. Aho] и др. *The Design and Analysis of Computer Algorithms*: «Граф $G(V,E)$ является конечным непустым множеством вершин V (они же — узлы) и ребер E . Если ребра E , которые имеют вид (a,b) , являются упорядоченными парами вершин, имеет место ориентированный граф. Если ребра E — неупорядоченные пары вершин, имеет место неориентированный граф». Да, вот так, и вы теперь наверняка думаете, что меня трудненько понять...

Hello world and Linux Format!

» Вывод простого файла Graphviz. Если вы сумеете его сгенерировать, вы готовы перейти к более сложным примерам.

```
"Hello world and Linux Format!";
}
```

Для такого простого примера не имеет никакого значения, ориентированный граф или неориентированный! Строка установки dpi (dpi = 300) необходима, чтобы получить приличное разрешение вывода — это никак не влияет на структуру графа. Для компиляции **simple.dot** надо выполнить следующую команду:

```
$ dot -Tpng simple.dot -o simple.png
```

Как видите, **dot** позволяет указать формат вывода, в данном случае — PNG. Другие поддерживаемые форматы файлов включают PS, PDF и SVG. (Внизу стр. 52 показан сгенерированный граф, имеющий только один узел.) Обратите внимание, что слово **digraph** означает ваше намерение создать ориентированный граф. Если вы хотите создать неориентированный граф, вместо этого надо использовать слово **graph**.

Если **dot** находит какие-либо синтаксические ошибки, вы получите сообщение об ошибке, подобное приведенному ниже:

```
$ dot -Tpng simple.dot -o simple.png
Error: simple.dot: syntax error in line 3 near 'digraph'
```

С процессом мы ознакомились; пора построить нечто более ценное, например хэш-таблицу. Возможность визуализации хэш-таблицы очень удобна и может быть особенно полезна, когда у вас есть сложная задача программирования, использующая хэш-таблицы. Соответствующий код Graphviz, сохраненный как **hashtable.dot**, таков:

```
digraph G
{
graph [ dpi = 144 ];
rankdir = LR;
node [shape=record, width=.2, height=.2];
node0 [label = "<p0> | <p1> | <p2> | <p3> | <p4> | |", height = 3];
node[ width=.2 ];
node1 [label = "{<e> r0 | 123 | <p> }" ];
node2 [label = "{<e> r10 | 13 | <p> }" ];
node3 [label = "{<e> r11 | 23 | <p> }" ];
node4 [label = "{<e> r12 | 326 | <p> }" ];
node5 [label = "{<e> r13 | 1f3 | <p> }" ];
node6 [label = "{<e> r20 | 143 | <p> }" ];
node7 [label = "{<e> r40 | b23 | <p> }" ];
node0:p0 -> node1:e;
node0:p1 -> node2:e;
node2:p -> node3:e;
node3:p -> node4:e;
node4:p -> node5:e;
node0:p2 -> node6:e;
node0:p4 -> node7:e;
}
```

(Вывод этого кода показан на рис. внизу стр. 52). Вы сможете увидеть таблицу, скомпилировав код с помощью

```
$ dot -T png hashtable.dot -o hashtable.png
```

Как видите, можно создать узел, поставив имя перед [], а ребро — поместив -> между двумя именами узлов, которые вы хотите соединить. Каждый узел может иметь заголовок, который добавляется с помощью атрибута **label**. Символ | разделяет различные части метки узла. Чтобы впоследствии получить к ним доступ, используйте имя узла, двоеточие (:) и идентификатор части метки, которую вы хотите использовать (например, **node4:p**). Имя каждой части метки определяется внутри тега <> (например, <e>). Такое обозначение позволяет более точно соединить два узла.

Каждый оператор **node[]**, который вы помещаете в свой код, влияет на все узлы в графе, начиная с позиции, в которой вы его размещаете. Команда **rankdir=LR** велит языку DOT размещать узлы слева направо, поскольку по умолчанию графы строятся сверху вниз.

Создание скриптов

Perl, как и любой другой язык программирования, может создавать простой код DOT без привлечения специального модуля. Следующий код Perl, сохраненный как **useGV.pl**, иллюстрирует использование Perl без каких-либо специальных модулей для создания простого кода Graphviz:

```
#!/usr/bin/perl -w
use strict;
die <<Thanatos unless @ARGV;
usage:
$0 directory
Thanatos
# Препамбула
print <<START;
digraph G
{
rankdir = LR;
graph [dpi = 300, bgcolor = "gray"];
nodesep=.05;
node[height=.05, shape=record, fontsize=10, color="red"];
START
my $counter = 0;
foreach my $arg (@ARGV)
{
print "\t", $arg, $counter, " [shape=box,label=\"$arg\"];";
print "\n";
$counter++;
}
# Создание ребер
```

Скорая помощь

Для получения более подробной информации о модуле Graphviz для Python 3 см. <https://pypi.python.org/pypi/graphviz>. Есть еще один пакет Python 3, называемый **pygraphviz** — (<https://pypi.python.org/pypi/pygraphviz>). Аналогично, для получения дополнительной информации о модуле Perl, называемом **Graphviz2**, перейдите на <http://bit.ly/GraphViz2>.

»

Использование C++

Библиотека Boost Graph, представляющая собой набор переносимых исходных библиотек C++, содержит интерфейс для языка DOT. Вы можете увидеть, например, код C++ для **graphViz.cpp**, который иллюстрирует создание кода Graphviz на C++. Упомянутый код пытается быть как можно проще, потому что C++ иногда бывает трудно понять, — он просто создает два узла и одно ребро, соединяющее эти два узла. Просто нужно помнить, что для генерации кода Graphviz вы можете использовать даже C++.

Компиляция и выполнение **graphViz.cpp** дает следующий вывод:

```
$ g++ graphViz.cpp -o graphViz
$ ./graphViz
$ ls -l cpp.dot
-rw-r--r-- 1 mtsouk mtsouk 49 Nov 2 10:04 cpp.dot
$ cat cpp.dot
digraph G {
0[label=one];
1[label=two];
```

```
0->1;
}
$ dot -Tpng cpp.dot -o cpp.png
$ ls -l cpp.png
-rw-r--r-- 1 mtsouk mtsouk 3849 Nov 2 10:05 cpp.png
```

Библиотека Boost Graph способна создавать графы и посложнее, для чего потребуются еще более хитроумные команды C++. Если вас заинтересовала библиотека Boost, можете начать с посещения www.boost.org.

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



➤ Скрипт `useGV.pl` демонстрирует, как создать код DOT без применения специальных модулей.

```

my $i = 0;
for ($i=0; $i< $#ARGV; $i++)
{
    print "\t"."$ARGV[$i]->$ARGV[($i+1)].($i+1).";
    if ($i == ($#ARGV-1))
    {
        last;
    }
}
# Закрывает файл Graphviz
print <<END;
}
END
  
```

Скрипт `useGV.pl` принимает несколько аргументов командной строки и создает с ними узлы: первый аргумент командной строки связан со вторым, второй — с третьим, и т.д. Выполнение `useGV.pl` дает вывод следующего вида, который надо обработать одним из инструментов Graphviz, чтобы получить желаемый график:

```

digraph G
{
    rankdir = LR;
    graph [dpi = 300, bgcolor = "gray"];
    nodesep=.05;
    node[height=.05, shape=record, fontsize=10, color="red"];
    a0 [shape=box, label="a"];
    b1 [shape=box, label="b"];
    c2 [shape=box, label="c"];
    a3 [shape=box, label="a"];
    a0->b1;
    b1->c2;
    c2->a3;
}
  
```

```

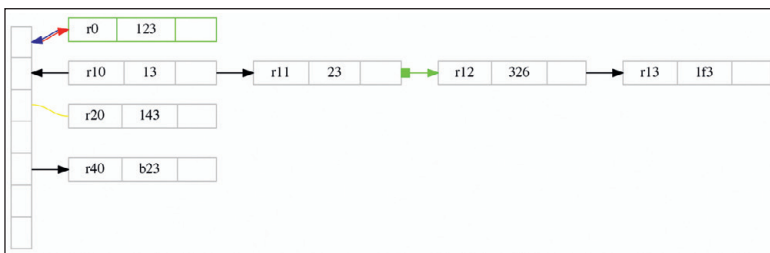
$ ./useGV.pl a b c a > t.dot
$ dot -Tpng t.dot -o t.png
  
```

Наведем красоту в выводе

Graphviz позволяет вносить изменения во внешний вид основного графа или его узлов. В этом разделе мы изменим внешний вид `hashtable.dot`. Однако фактическая информация, а также количество узлов останутся теми же. Новый файл `.dot` будет называться `beautifyHT.dot`. Следующий вывод показывает сравнение между `hashtable.dot` и `beautifyHT.dot` с помощью утилиты командной строки `diff`:

```

$ diff hashtable.dot beautifyHT.dot
5c5
  
```



➤ Это вывод `beautifyHT.dot`, модифицированной версии `hashtable.dot`, где можно увидеть, как изменять различные атрибуты узла или ребра.

```

< node [shape=record, width=.2, height=.2];
---
> node [shape=record, width=.2, height=.2, color="gray"];9c9
< node1 [label = "{<e> r0 | 123 | <p>}"];
---
> node1 [label = "{<e> r0 | 123 | <p>}", color="green"];17,18c17,18
< node0:p0 -> node1:e;
< node0:p1 -> node2:e;
---
> node0:p0 -> node1:e [dir=both color="red:blue"];
> node0:p1 -> node2:e [dir=back arrowhead=diamond];20,22c20,22
< node3:p -> node4:e;
< node4:p -> node5:e;
< node0:p2 -> node6:e;
---
> node3:p -> node4:e [dir=both arrowtail=box color="green"];
> node4:p -> node5:e [dir=forward];
> node0:p2 -> node6:e [dir=none color="yellow"];
  
```

Посмотрите на снимок графа, созданного `beautifyHT.dot` (внизу), чтобы лучше понять значение внешнего вида созданной хэш-таблицы.

Атрибут `color` изменяет цвет узла, тогда как атрибут `shape` изменяет форму узла. Кроме того, атрибут `dir`, применяемый к ребрам, определяет, будет ли ребро иметь две стрелки, одну или ни одной. Стиль стрелки можно вдобавок задать атрибутами `arrowhead` и `arrowtail`. Наводя красоту в выводе, постарайтесь не переусердствовать, поскольку изображение, рассеивающее внимание, не служит своей цели. Полный список атрибутов узла, ребра и графа можно найти на <http://www.graphviz.org/doc/info/attrs.html>.

Скрипт на Python

В этом разделе мы представим скрипт на Python 3, генерирующий код DOT с помощью модуля. Скрипт с именем `dirs.py` рисует структуру каталогов, начиная с заданного каталога. Прежде чем продолжить, надо установить пакет Python 3, с помощью

```
$ pip3 install graphviz.
```

Теперь пакет Python 3 готов к употреблению. Прежде чем перейти к более сложному примеру, давайте представим простой пример Python 3, чтобы лучше понять, как работает модуль Graphviz:

```

#!/usr/bin/env python3
from graphviz import Digraph

dot = Digraph('Using graphviz module!')
dot.node_attr.update(color='lightblue', style='filled')
# Добавим две вершины
dot.node('Linux')
dot.node('Format')
# Создадим ребро
dot.edge('Linux', 'Format')
dot.render("pythonSimple.dot", view=False)
  
```

Как видите, этот код достаточно прост для понимания. Создание всех узлов и ребер выполняется модулем Graphviz от Python 3, так что вам не нужно самим писать никакого кода Graphviz. Выполнение этого примера, сохраненного как `pythonSimple.py`, создает такой файл:

```

$ ls -l pythonSimple.*
-rw-r--r-- 1 mtsouk mtsouk 109 Nov 2 08:30 pythonSimple.dot
-rw-r--r-- 1 mtsouk mtsouk 12337 Nov 2 08:30 pythonSimple.pdf
-rwxr-xr-x 1 mtsouk mtsouk 294 Nov 2 08:30 pythonSimple.py
  
```

Хорошо, что модуль Graphviz автоматически сохраняет сгенерированный файл DOT, а значит, впоследствии вы, если захотите,

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

Утилита dot2tex

LaTeX является превосходным инструментом для изготовления верстки документов и книг; его используют многие студенты, исследователи и профессионалы. Поэтому возможность перевести ваш вывод Graphviz в LaTeX очень удобна, поскольку инструменты Unix не являются изолированными объектами.

Утилита `dot2tex` делает именно это! В системе Debian вы можете установить ее следующим образом:

```
# apt-get install dot2tex
```

А вот команда, показывающая, как использовать `dot2tex` для преобразования нашего файла примера `simple.dot` в формат LaTeX:

```
$ dot -Txdot simple.dot | dot2tex > simple.tex
```

Эта команда состоит из двух частей. Первая часть — выполнение команды `dot`; здесь `dot` генерирует файл, подходящий для утилиты `dot2tex`, который поступает на вход утилиты `dot2tex`, используя канал Unix (вторая часть). Вывод `dot2tex` сохраняется в новый файл с именем `simple.tex`, который содержит уже код LaTeX. Чтобы скомпилировать

`simple.tex` и создать файл PDF, можно использовать утилиту `pdflatex`:

```
$ pdflatex simple.tex
```

```
...
```

```
Output written on simple.pdf (1 page, 14049 bytes) [Вывод записан в simple.pdf (1 страница, 14049 байт)].
Transcript written on simple.log [Транскрипт записан в simple.log].
```

Утилита `dot2tex` имеет много параметров, о которых можно узнать на <https://github.com/kjellmf/dot2tex>.

можете его подправить — имя файла указано как первый аргумент метода `dot.render()`.

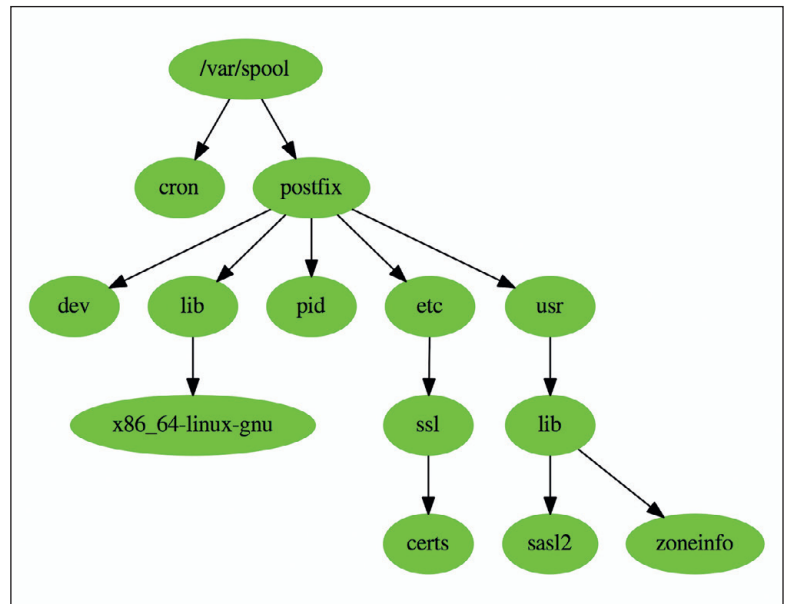
Теперь перейдем к самому скрипту Python 3. Код `dirs.py`, требующий один аргумент командной строки, который должен быть каталогом, таков:

```
#!/usr/bin/env python3
import os
import sys
from graphviz import Digraph

if len(sys.argv) == 2:
    directory = str(sys.argv[1])
else:
    print('Usage: ', sys.argv[0], 'directory')
    sys.exit(0)
if os.path.isdir(directory) == False:
    print(directory, 'not a directory!')
    sys.exit(0)
dot = Digraph('A Directory Structure')
dot.node_attr.update(color='green', style='filled')
for root, dirs, files in os.walk(directory):
    print(root)
    if root == directory:
        dot.node(directory)
        continue
    if os.path.isdir(root):
        # Левая вершина
        left = os.path.dirname(os.path.normpath((root)))
        # Правая вершина
        right = os.path.basename(os.path.normpath((root)))
        dot.node(left)
        dot.node(right)
        dot.edge(left, right)
dot.render("directory.dot", view=False)
```

Скрипт `dirs.py` использует метод `os.walk()` для посещения всех необходимых каталогов и создания требуемого графа. Сложная часть в `dirs.py` там, где вы создаете ребра между узлами — и снова помогает метод `os.walk()`, потому что при каждом изменении текущего каталога вы знаете, что вам нужно создать новое ребро.

Первым аргументом метода `node()` является внутреннее представление узла, а вторым аргументом — его метка! Каждый узел представлен в виде полного пути каталога, что упрощает работу и позволяет избежать дублирования соединений, поскольку полный путь уникален среди всех каталогов одной машины. Метка каждого узла является его базовым именем, а базовое имя пути — это последняя часть пути, которая также является каталогом. Обратите внимание, что два разных пути могут иметь одно и то же базовое имя: базовое имя как `/home/mtsouk`, так и `/users/mtsouk` — это `mtsouk`.



Причина жесткого кодирования имени файла, сгенерированного `dot`, в том, чтобы избежать сюрпризов с путями каталогов, содержащих внутри `.` или `..`:

```
$ ./dirs.py /var/spool
/var/spool
/var/spool/cron
/var/spool/postfix
/var/spool/postfix/dev
/var/spool/postfix/lib
/var/spool/postfix/lib/x86_64-linux-gnu
/var/spool/postfix/pid
/var/spool/postfix/etc
/var/spool/postfix/etc/ssl
/var/spool/postfix/etc/ssl/certs
/var/spool/postfix/usr
/var/spool/postfix/usr/lib
/var/spool/postfix/usr/lib/sas12
/var/spool/postfix/usr/lib/zoneinfo
```

(Графический вывод — на рис. вверху.) Попытавшись обработать файл другого типа вместо каталога, вы получите следующий результат:

```
$ ./dirs.py dirs.py
dirs.py not a directory!
```

После прочтения этого руководства важно помнить, что Graphviz выполняет за вас всю грязную работу, чтобы вы могли сосредоточиться на значимых вещах — фактической информации, содержащейся в графе. **LXF**

➤ Это вывод скрипта `dirs.py` на Python 3, который строит структуру каталогов, начиная с заданного.

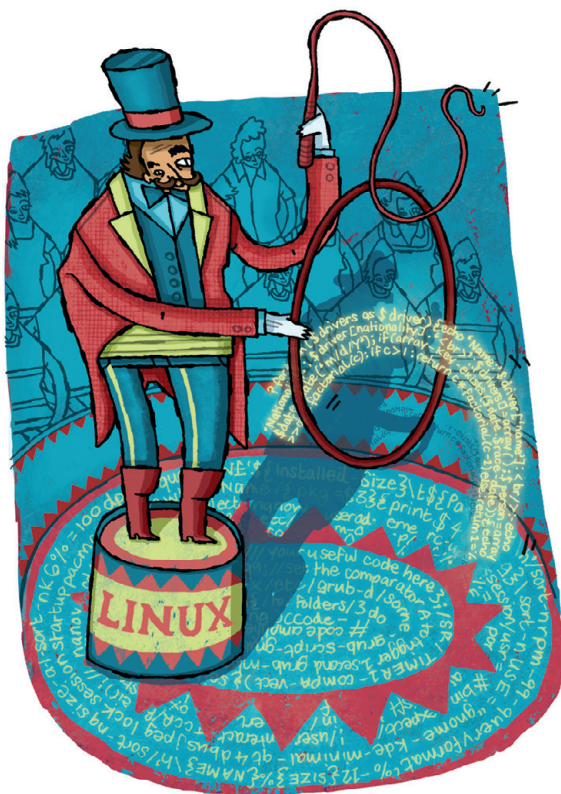
HAProxy: TCP-балансировка

Михалис Цукалос научит вас, как установить и настроить HAProxy для перераспределения нагрузки MySQL Replica Set, успев все это до обеда.



Наш эксперт

Михалис Цукалос — администратор баз данных и Unix, программист и математик, который любит писать статьи и узнавать что-то новое.



```
2. mtsouk@LTng: /etc/haproxy (ssh)
mtsouk@LTng: /etc/haproxy$ cat haproxy.cfg.orig
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # Default ciphers to use on SSL-enabled listening sockets.
    # For more information, see ciphers(1SSL). This list is from:
    # https://nynek.me/articles/hardening-your-web-servers-ssl-ciphers/
    ssl-default-bind-ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+
    AES:ECDH+3DES:DH+3DES:RSA+AESGCM:RSA+AES:RSA+3DES:!aNULL:!MD5:!DSS
    ssl-default-bind-options no-sslv3

defaults
    log global
    mode http
    option httplog
    option dontlognull
    timeout connect 5000
    timeout client 50000
    timeout server 50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
```

» Содержимое файла настройки HAProxy по умолчанию /etc/haproxy/errors/400.http.

Затем вы можете узнать используемую вами версию HAProxy следующим образом:

```
# haproxy -v
HA-Proxy version 1.5.8 2014/10/31
Copyright 2000-2014 Willy Tarreau w@1wt.eu
```

Пожалуйста, помните, что стабильные дистрибутивы Debian имеют тенденцию устанавливать более старые версии пакетов, поскольку те лучше защищены и более стабильны, несмотря на тот факт, что у них меньше функций. На момент написания этой статьи последними стабильными версиями HAProxy являются 1.7.2, 1.6.11 и 1.5.19. Если вашей главной заботой является стабильность, используйте версию 1.6.x или 1.5.x, поскольку версия 1.7.x HAProxy достаточно новая.

Настройка

Главной директорией настройки HAProxy является /etc/haproxy, где содержится следующее:

```
# ls -l /etc/haproxy
total 8
drwxr-xr-x 2 root root 4096 Jan 21 19:16 errors
-rw-r--r-- 1 root root 1129 Jul 14 2015 haproxy.cfg
# ls /etc/haproxy/errors
400.http 403.http 408.http 500.http 502.http 503.http 504.http
```

Главным файлом настройки HAProxy является /etc/haproxy/haproxy.cfg. Директория ошибок [errors] содержит различные сообщения об ошибках, относящиеся к данным кодам состояния HTTP.

HAProxy (что означает High Availability Proxy — прокси с высокой доступностью) — это балансировщик нагрузки обратного прокси для протоколов TCP и HTTP. На нашем уроке мы будем использовать HAProxy вместе с MySQL, чтобы продемонстрировать возможности перераспределения нагрузки. Однако HAProxy пригоден для работы и с другими TCP-серверами, такими как Apache и Nginx.

Пожалуйста, делайте резервные копии каждого файла, в который вносите изменения, чтобы было проще вернуться к изначальной настройке. Основная причина тут в том, что программы, работающие с сайтами и серверами баз данных, при неправильной настройке могут сделать их недоступными для клиентов, из-за чего потребуются это быстро исправить.

Установка

Установка HAProxy на компьютеры с Debian или Ubuntu Linux не сложнее, чем выполнение следующей команды с правами root:

```
# apt-get install haproxy
```

Скорая помощь

Надо ли вам использовать HAProxy? Это прекрасный инструмент для перераспределения нагрузки, в основном благодаря простоте и прозрачности его операций. Вам, определенно, стоит рассмотреть возможность его применения.

hatox & haproxyctl

hatox — это интерактивный клиент на *ncurses* для HAProxy, а *haproxyctl* — утилита для управления HAProxy из командной строки. Ни один из них для работы HAProxy не требуется, но если вы их установите, они могут облегчить вашу жизнь. В системах Debian и Ubuntu Linux вы можете установить их так:

```
# apt-get install hatox haproxyctl
```

haproxyctl позволяет быстро управлять и просматривать HAProxy с помощью своих удобных опций командной строки:

```
$ sudo haproxyctl show health
$ sudo haproxyctl show stat
```

Первая команда позволяет бегло проверить статус HAProxy, а вторая возвращает счетчики

для каждого прокси и сервера. Если вы захотите узнать о *haproxyctl* больше, можете почитать его map-страницу.

Изображение на стр. 59 демонстрирует утилиту *hatox* в действии. Чтобы обеспечить работу *hatox*, вам необходимо будет активировать сокет HAProxy Unix.

Скриншот выше показывает содержимое `/etc/haproxy/haproxy.cfg` и `/etc/haproxy/errors/400.http`.

Продолжим...

Базовая установка закончена; прежде чем продолжить демонстрацию применения HAProxy для распределения нагрузки нескольких экземпляров *MySQL*, давайте взглянем на основную функцию. HAProxy запускается с помощью

```
# service haproxy start
Следующий вывод подтверждает, что HAProxy запущен и успешно работает:
# ps ax | grep -i haproxy
17747 ? Ss 0:00 /usr/sbin/haproxy-systemd-wrapper -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid
17749 ? S 0:00 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -Ds
17751 ? Ss 0:00 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -Ds
```

Приятно то, что предыдущий вывод показывает различные полезные вещи о запущенном экземпляре HAProxy, включая полный путь используемого файла настройки и место, где вы можете найти ID его процесса (`/run/haproxy.pid`).

Аналогично можно остановить HAProxy:

```
# service haproxy stop
```

Теперь пора распределить нагрузку двух экземпляров *MySQL*. Для целей нашего урока два экземпляра *MySQL* будут в одной сети использовать два тестовых компьютера — ведь не пробовать же такие вещи на рабочем сервере! IP-адрес первого компьютера — `192.168.1.200` (MyA), а IP-адрес второго компьютера будет `192.168.1.4` (MyB). IP-адрес компьютера HAProxy — `10.0.2.15`. Как видите, он принадлежит другой сети, но это не должно быть проблемой, пока две сети могут успешно взаимодействовать друг с другом. Чтобы лучше понимать происходящее, помните, что HAProxy работает на Виртуальной Машине [Virtual Machine] на MyB.

Пожалуйста, учтите, что вам, возможно, понадобится включить сетевой доступ к обоим серверам *MySQL*, позволив им слушать подключения TCP/IP — эта функция отключена по умолчанию из соображений безопасности. Следующий вывод `diff` показывает изменение, которое надо внести в файл настройки *MySQL*, чтобы активировать удаленные подключения TCP/IP:

```
$ diff my.cnf my.cnf.orig
31d30
< bind-address = 192.168.1.200
```

Оба процесса *MySQL* по умолчанию слушают порт *MySQL*, его номер — 3306. Затем вам понадобится предоставить требуемые разрешения, включающие удаленный доступ для выбранных пользователей, которым в нашем случае является `root`:

```
mysql> GRANT ALL ON *.* TO 'root'@'192.168.1.200' IDENTIFIED BY 'mypass';
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

Вам также следует выполнить следующую команду на MyB:

```
mysql> GRANT ALL ON *.* TO 'root'@'192.168.1.4' IDENTIFIED BY 'mypass';
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

Не забудьте, внося эти изменения, перезагрузить оба сервера *MySQL*. Вы можете убедиться, что оба экземпляра *MySQL* запущены и работают, и к ним можно получить доступ из сети, попытавшись подключиться к каждому с третьего компьютера следующим образом:

```
ifconfig | grep "inet addr" | head -1
inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
$ mysql -u root -h 192.168.1.4 -p
$ mysql -u root -h 192.168.1.200 -p
```

Вам также понадобится настроить в *MySQL* Master-Master Replication. Это выходит за рамки нашего урока, поэтому, прежде чем продолжить, просто убедитесь, что репликация работает должным образом; простейшим способом сделать это является создание новой таблицы в новой базе данных в одном из экземпляров *MySQL* и проверка, скопировалась ли она в другую базу данных.

И наконец, надо будет создать в каждой базе данных *MySQL* еще двух пользователей *MySQL*, чтобы позволить HAProxy отслеживать их — хорошая новость в том, что если репликация работает как полагается, вам понадобится выполнить следующую команду один-единственный раз:

```
$ mysql -u root -p
mysql> INSERT INTO mysql.user (Host,User) values ('10.0.2.15','haproxy_check'); FLUSH PRIVILEGES;
mysql> GRANT ALL PRIVILEGES ON *.* TO 'haproxy_root'@'10.0.2.15' IDENTIFIED BY 'password' WITH GRANT OPTION; FLUSH PRIVILEGES;
```



HAProxy поддерживает только TCP-протокол. Если вам нужен протокол UDP, можете взглянуть на такие программы, как *udp-balancer*, *udpbalancer* и, угадайте-ка, *Nginx*! Вы можете найти больше информации об *udp-balancer* и *udpbalancer* на <https://github.com/vadimmedes/udp-balancer> и <https://github.com/johann8384/udpbalancer>.

```
2. mtsouk@LTTng: /etc/haproxy (ssh)
Jan 31 22:43:17 LTTng haproxy-systemd-wrapper[7335]: haproxy-systemd-wrapper: SIGINT -> 733
Jan 31 22:43:17 LTTng haproxy-systemd-wrapper[7335]: haproxy-systemd-wrapper: exit, haproxy
RC=0
Jan 31 22:43:17 LTTng haproxy-systemd-wrapper[7391]: haproxy-systemd-wrapper: executing /us
r/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -Ds
Jan 31 22:43:17 LTTng haproxy-systemd-wrapper[7391]: [WARNING] 030/224317 (7392) : parsing
/etc/haproxy/haproxy.cfg:24 : 'option httplog' not usable with proxy 'mysql-setup' (needs
'mode http'). Falling back to 'option tcplog'.
Jan 31 22:43:17 LTTng haproxy-systemd-wrapper[7391]: [WARNING] 030/224317 (7392) : config :
proxy '127.0.0.1:8080' has no 'bind' directive. Please declare it as a backend if this was
intended.
Jan 31 22:43:17 LTTng haproxy[7392]: Proxy mysql-setup started.
Jan 31 22:43:17 LTTng haproxy[7392]: Proxy mysql-setup started.
Jan 31 22:43:17 LTTng haproxy[7392]: Proxy 127.0.0.1:8080 started.
Jan 31 22:43:17 LTTng haproxy[7392]: Proxy 127.0.0.1:8080 started.
Jan 31 22:46:23 LTTng haproxy-systemd-wrapper[7391]: haproxy-systemd-wrapper: SIGINT -> 739
3
Jan 31 22:46:23 LTTng haproxy-systemd-wrapper[7391]: haproxy-systemd-wrapper: exit, haproxy
RC=0
Jan 31 22:46:23 LTTng haproxy-systemd-wrapper[9006]: haproxy-systemd-wrapper: executing /us
r/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -Ds
```

» Информация, которую следует ожидать обнаружить в файлах логов HAProxy.

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

```

2. mtsouk@LTTng: /etc/haproxy (ssh)
mtsouk@LTTng: /etc/haproxy$ sudo nc -U /run/haproxy/admin.sock
prompt

> show info
Name: HAProxy
Version: 1.6.3
Release_date: 2015/12/25
Nproc: 1
Process_num: 1
Pid: 9541
Uptime: 0d 0h01m35s
Uptime_sec: 95
Memmax_MB: 0
Ulimit-n: 4033
Maxsock: 4033
Maxconn: 2000
Hard_maxconn: 2000
CurrConns: 0
CumConns: 2
CumReq: 2
MaxSslConns: 0
CurrSslConns: 0
CumSslConns: 0
Maxpipes: 0
PipesUsed: 0
PipesFree: 0
ConnRate: 0
    
```

➤ Как использовать сокет HAProxy Unix, которым в данном случае является /run/haproxy/admin.sock, для чтения показателей.



Надо ли вам гнаться за само свежей версией HAProxy? Использование последней версии программы, столь критичной для вашей инфраструктуры в качестве прокси-сервера и балансировщика нагрузки, не всегда является лучшей идеей, особенно когда вам не нужны ее новые функции. Если она работает, не трогайте ее.

Пожалуйста, учтите, что вам понадобится IP-адрес компьютера, на котором запущен сервер HAProxy. Выполнение тех же команд с IP-адресами двух MySQL-серверов может облегчить вашу жизнь, особенно если вы используете любую из виртуальных машин, так что запустите их.

Затем надо внести изменения в файл настройки HAProxy. Пожалуйста, держите в голове, что изначальный файл haproxy.cfg сохранен как haproxy.cfg.orig. Следующий вывод, сделанный с помощью утилиты командной строки diff, показывает изменения в изначальном файле haproxy.cfg:

```

$ diff haproxy.cfg haproxy.cfg.orig
36,44d35
<
< listen mysql-setup
< bind 127.0.0.1:3306
< mode tcp
< option mysql-check user haproxy_check
< balance roundrobin
< server mysql-1 192.168.1.4:3306 check
< server mysql-2 192.168.1.200:3306 check
    
```

Report for HAProxy - Mozilla Firefox

Statistics Report for pid 11494

General process information

mysql-setup

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Status	LastChk	Weight	Act	Bck	Chk	Dwn	Downtime					
Cur	Max	Limit	Cur	Max	Limit	Total	LibTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	OPEN	
Frontend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mysql-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mysql-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Backend	0	0	0	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0

➤ Web-страница статистики Statistics HAProxy после успешной настройки HAProxy.

Использование IP-адресов вместо имен компьютеров, возможно, в какой-то степени усложнит вашу жизнь, зато уберекет HAProxy от необходимости разрешать имена компьютеров.

Пожалуйста, убедитесь, что на компьютере с работающим HAProxy не запущен MySQL в любом виде, поскольку в этом случае следующие команды попытаются подключиться к локальной копии MySQL:

```

$ mysql -h 127.0.0.1 -u haproxy_root -p -e "show variables like 'server_id'"
Enter password:
+-----+-----+
| Variable_name | Value |
+-----+-----+
| server_id | 2 |
+-----+-----+

$ mysql -h 127.0.0.1 -u haproxy_root -p -e "show variables like 'server_id'"
Enter password:
+-----+-----+
| Variable_name | Value |
+-----+-----+
| server_id | 1 |
+-----+-----+
    
```

Поскольку у каждой репликации MySQL свой server_id, предыдущий вывод сообщает, что вы можете подключиться к обоим серверам репликации MySQL при запросе localhost — компьютера, на котором запущен HAProxy. Итак, отныне, когда вам понадобится MySQL-сервер, вы можете вместо IP-адреса одного из двух MySQL-серверов дать только IP-адрес и номер порта сервера HAProxy, а HAProxy позаботится об остальном. Это прозрачный способ использования сервисов без того, чтобы мир узнал всю подноготную.

Лог-файлы

В системе Ubuntu файлы журналов HAProxy можно найти в /var/log/haproxy.log. Записи, которые вы обнаружите внутри /var/log/haproxy.log, будут похожи на следующие:

```

Jan 31 23:19:08 LTTng haproxy[936]: Server mysql-setup/mysql-1 is DOWN, reason: Layer4 connection problem, info: "General socket error (Network is unreachable)", check duration: 0ms.
Jan 31 23:21:57 LTTng haproxy[936]: 127.0.0.1:52778 [31/Jan/2017:23:21:57.856] stats stats<STATS> 0/0/0/0 200 1346 -- LR-- 1/1/0/0/0 0/0 "GET /hastats:csv HTTP/1.1"
    
```

Скриншот на стр. 57 показывает больше записей из /var/log/haproxy.log, чтобы создать лучшее понимание информации в этом файле. Основная идея в том, что, изучая новый аспект программы, надо следить за лог-файлами, поскольку лог-файлы дают лучшее понимание того, что происходит за кулисами.

HAProxy предлагает большое количество показателей метрики, позволяя отслеживать то, как он работает, а также его производительность. Эти показатели делятся на три основных категории: интерфейс [frontend], движок [backend] и показатели здоровья [health metrics]. Показатели интерфейса собирают информацию о клиентах, а показатели движка собирают данные о доступности и статусе серверных машин. Последняя разновидность показателей сообщает вам о статусе HAProxy.

Показатели интерфейса включают такую информацию, как количество HTTP-запросов за секунду (req_rate), количество ошибок запросов (ereq) и количество отправленных байтов (bout). Показатели движка включают способы измерить среднее время отклика (rtime) и количество запросов вне очереди (qcur).

Самый безопасный способ получить показатели HAProxy — использовать сокеты UNIX. В ОС Ubuntu Linux HAProxy по умолчанию

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

HAProxy против Nginx

Nginx — это не просто web-сервер; он может с тем же успехом заниматься многими другими задачами, включая некоторые задачи HAProxy. Основным преимуществом *Nginx* над HAProxy является предлагаемая им интеграция, и к тому же вам понадобится изучить только одну программу, чтобы сделать свою работу.

С другой стороны, это является и недостатком, поскольку всё, что вы делаете, зависит от единственной программы. Вдобавок, в большинстве своем, специализированные программы склонны лучше справляться со своими задачами,

чем относительно многофункциональные инструменты: в них меньше ошибок, поскольку их реализация короче, и они имеют лучшую производительность благодаря более простому дизайну.

Конечно, интеграция *Nginx* позволяет осуществлять маршрутизацию на основе информации, обнаруженной в среде HTTP, что включает пути URL и cookie — это также означает, что *Nginx* может выполнять сложные задачи с большей легкостью. И, наконец, у *Nginx* имеется встроенная поддержка SSL, и он также является кэширующим сервером. Тем не менее, HAProxy проще в установке,

настройке и использовании, и его работа прозрачна, а значит, после должной настройки вы вообще забудете о существовании HAProxy. Помимо всех этих преимуществ, HAProxy работает с TCP-сервисами вообще, тогда как *Nginx* работает только с HTTP. Другим преимуществом HAProxy является то, что он заново заходит на «сдохший» сервер практически сразу после перезапуска, а *Nginx* некоторое время ждет.

Основной совет — использовать *Nginx*, когда вам нужен web-сервер; в ином случае, использовать HAProxy будет мудрее.

имеет поддержку для желаемого сокета Unix, так что вам не надо будет делать что-то еще. Чтобы убедиться, что сокет Unix (`/run/haproxy/admin.sock`) был создан и может использоваться, можно сделать следующее:

```
$ sudo nc -U /run/haproxy/admin.sock
prompt
> show info
...
```

Рис. вверху стр. 58 также показывает вывод, который можно получить из сокета Unix, используя утилиту командной строки *netcat*. Помните: если вы собираетесь делать с HAProxy нечто серьезное, надо научиться интерпретировать его показатели.

Страница мониторинга

HAProxy предлагает страницу мониторинга, где более подробные сведения о работе HAProxy представлены в графическом виде. Именно ее вы должны первым делом посетить при проблемах с HAProxy. Однако эта страница не включена по умолчанию. Чтобы включить ее, вам следует добавить следующий блок в файл настройки HAProxy:

```
listen stats
bind *:8080
stats enable
stats hide-version
stats realm Haproxy\ Statistics
stats uri /hastens
stats auth user:password
```

Последняя строка позволяет задать действительные комбинации имени пользователя и пароля, а определение `uri` задает URL страницы статистики. Значение `bind` определяет номер порта, который будет слушать страница статистики. Чтобы изменения вступили в силу, понадобится перезапустить HAProxy.

Рис. внизу стр. 58 показывает страницу мониторинга HAProxy, которая обычно слушает порт 6427 адреса localhost, а значит, по умолчанию эта страница не доступна через Интернет. Однако представленная настройка использует номер порта 8080. Как видно по web-странице, можно также получить вывод в формате CSV, что бывает очень удобно, поскольку позволяет легко хранить значения показателей на сервере базы данных. Если вы используете HAProxy, страницу мониторинга вам, вероятнее всего, понадобится включить.

Пожалуйста, учтите, что HAProxy предоставляет свой собственный web-сервер для отображения страницы мониторинга.

Алгоритмы

HAProxy умеет использовать множество алгоритмов, чтобы решить, какой сервер будет выбран при балансировке нагрузки нескольких серверов. Используемые алгоритмы включают Round

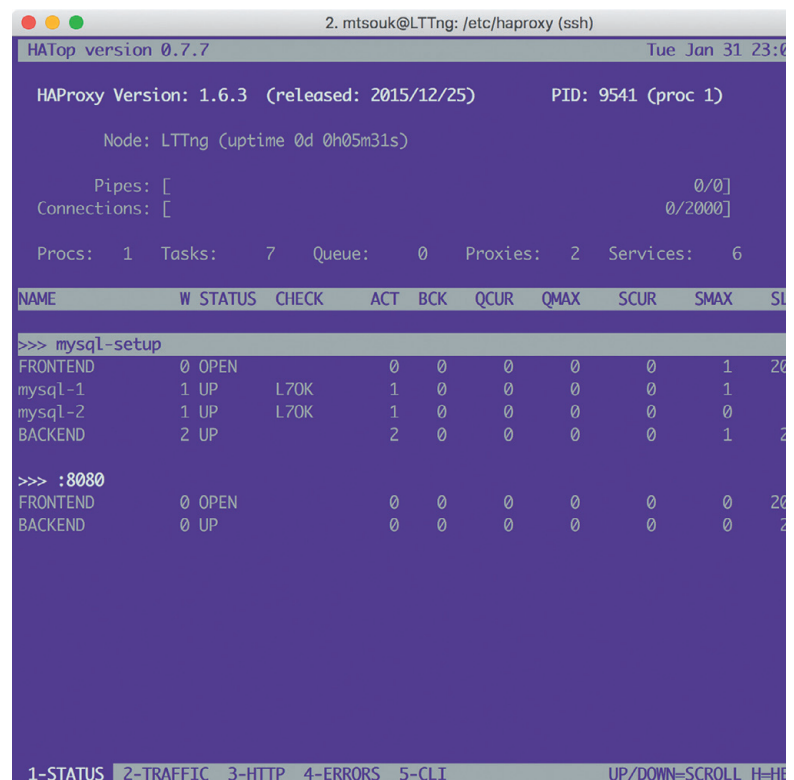
Robin — алгоритм, выбирающий сервер с наименьшим числом подключений, и другой, основанный на IP-адресе клиента. Второй метод обеспечивает подключение каждого IP-адреса всегда к одному и тому же серверу. Вдобавок вы можете приписать серверам вес, определяющие, как часто каждый из серверов будет выбираться по сравнению с другими.

Если у вас нет проблем с производительностью, алгоритмы HAProxy лучше не трогать.

Новые функции

Удобная функция последней версии HAProxy — если файл настройки, заданный с помощью ключа `-f`, является директорией, то все файлы в директории будут загружены в алфавитном порядке. Вдобавок в ней есть поддержка OpenSSL 1.1.0, производительность лучше, чем в 1.6.x, и исправлены многие ошибки.

Вы можете найти больше информации о HAProxy на <http://www.haproxy.org>, а полную документацию всех версий HAProxy см. на <http://cbonte.github.io/haproxy-dconv>. **LXF**



► Утилита *hatop* в действии. *Hatop* использует сокет HAProxy Unix, и ее вывод похож на вывод утилиты *top*.

GnuPG: Знайте, кому доверять

Джон Лэйн исследует запутанный мир удостоверения подлинности ключей PGP.



Наш эксперт

Джон Лэйн доверяет той точке зрения, что GnuPG доверяет ему доверять ключу, который доверяет сети доверия...

PGP означает “Pretty Good Privacy [Достаточно Хорошая Конфиденциальность]”. То есть, не идеальный, но, вероятно, вполне приличный способ шифровать сообщения и другие документы или удостоверять, что они не были изменены кем-то другим. Если вы поймете, что такое его «Сеть Доверия [Web of Trust]», дело в шляпе. Многие пользователи Linux в какой-то момент столкнутся с Gnu Privacy Guard (*GnuPG*), реализацией стандарта OpenPGP, но чтобы использовать его по максимуму, необходимо распутать эту паутину.

Вы используете чей-то ключ, чтобы отправить этому лицу зашифрованное сообщение или документ, или чтобы подтвердить подлинность подписанного сообщения или документа, отправленного вам. Но создать ключ от чьего-то лица может любой, и прежде чем использовать ключ, надо убедиться, что его создатель действительно является тем, кем себя называет, и что его личность правильно представлена. Тут-то и начинаются сложности. Вы можете сделать всё это сами или поручить это другим.

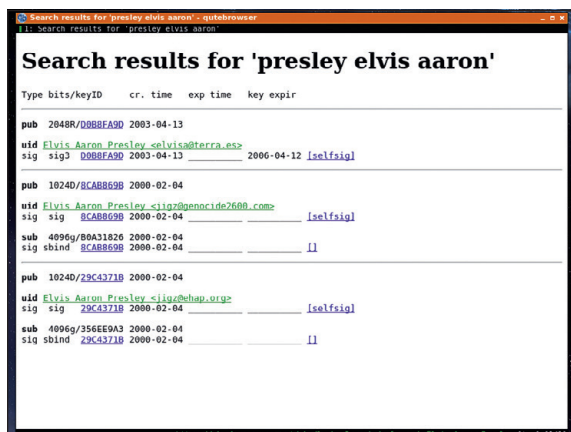
OpenPGP определяет модель доверия под названием Web of Trust (WoT), которую обычно применяет *GnuPG*. OpenPGP также поддерживает альтернативные модели, включая простое доверие и Trust On First Use (TOFU) [Доверие на первый раз]; однако основным принципом доверия в OpenPGP является WoT. К сожалению, понять эту модель подтверждения, доверия и удостоверения бывает трудно.

На нашем уроке мы постигнем черную магию подтверждения подлинности ключей и объясним, что значит подтвердить подлинность, подписать и удостоверить ключ, или доверять ключу или его владельцу. Ключами в данном контексте являются открытые (они же — публичные) ключи OpenPGP (мы предполагаем наличие у вас представления о *GnuPG* и об асимметричной криптографии с открытым ключом). Подтверждение, доверие и подлинность относятся к открытым, а не к частным (они же — приватные) ключам, поэтому когда в данном руководстве используется слово «ключ», подразумевается открытый ключ.

Лучший способ получить чей-то ключ — личная встреча, чтобы файл ключа, или USB-брелок, или CD-ROM передали вам в руки. Тогда вы можете быть уверены в его подлинности. На практике люди могут использовать другие способы — например, публикацию ключа на своем сайте или на сервере открытых ключей. Это сервисы в Интернете, некоторые из них основаны на web, и на них вы можете отправить запрос для получения ключей других людей. Однако поскольку любой может создать и загрузить ключи, неразумно доверять их подлинности, сначала в ней не удостоверившись.

Элвис покинул здание

Чтобы проиллюстрировать этот момент вы можете заглянуть на один из серверов ключей OpenPGP в Web. Направьте свой браузер на <https://sks-keyserver.net/> и задайте поиск “Elvis Aaron Presley”. Ключ найдется, причем не один — невзирая на тот факт,



» Не все ключи подлинны. Подделать ключ может любой. Чему доверять — решать вам.

что их предполагаемый владелец умер за несколько лет до появления на свет самой первой версии PGP. Если хотите нечто более современное, поищите “Linus Torvalds” — конечно, часть из них будут подлинными, но некоторые — определенно нет; зато они служат неплохим пояснением, почему надо проверять ключи, которыми вы пользуетесь! Вам надо уверовать в подлинность ключа до того, как вы ему доверитесь, поэтому самый основной принцип доверия — удостоверение подлинности. Это настоящее расследование, когда вы предпринимаете шаги, которые считаете нужными для подтверждения подлинности ключа посредством удостоверения его владельца и его личности. Обратите внимание, что это субъективное определение и, следовательно, для вас оно будет личным. У разных людей бывают разные подходы, от случайной догадки до настоятельных просьб о личной встрече и физической проверки официальных документов, подтверждающих личность, например, паспорта или иных официальных документов (часто используется водительское удостоверение). Вы можете считать ключ подлинным, только если вы уверены, что он действительно принадлежит предполагаемому владельцу, и знаете, что его владелец действительно является тем, за кого себя выдает.

Чтобы подтвердить подлинность ключа, вы его подписываете, после чего можете его использовать, поскольку *GnuPG* доверяет вашему подходу к подтверждению подлинности и, таким образом, принимает, что подписанные вами ключи пригодны.

Мы говорим «подписать» ключ, но процесс удостоверения также привязан к “UID” — id пользователя, которым может быть триада имя, адрес электронной почты и комментарий или, возможно, фотография. Удостоверение остается действительным, пока не будет отозван ключ и/или связанный с ним UID, даже при наличии иных изменений, однако для каждого UID нужны отдельные удостоверения.



Вы можете сделать подписи “local”, чтобы не разрешить их экспорт. В `man gpg` поищите “Isign”.

Песочница GnuPG

Желая опробовать наши примеры, не затрагивая своей собственной конфигурации *GnuPG*, вы можете создать отдельную конфигурацию *GnuPG* — либо добавив к каждой команде *gpg* опцию `--homedir`, либо настроив `GNUPGHOME`.

И то, и другое указывает на директорию, где *GnuPG* должен держать свои настройки, в том числе и вашу связку ключей в песочнице. Если выбрать последнюю опцию, то новая песочница создается так:

```
$ mkdir -m 700 $GNUPGHOME
$ gpg --gen-key
```

Помните, что для доступа к вашей реальной настройке вам надо сменить настройку `GNUPGHOME` или использовать `--homedir=~/gnupg`.

Если у ключа несколько значений UID, то *GnuPG* предупредит вас и спросит, хотите ли вы подписать их все сразу. Если не хотите, то вы выделяете и подписываете их индивидуально. В приведенных далее примерах значение UID только одно.

GnuPG поддерживает вашу связку ключей, где хранит собранные вами ключи и все удостоверения к ним. Ваша связка ключей представляет вашу собственную Сеть Доверия.

Чтобы продемонстрировать удостоверение, представим себе, что мы получили зашифрованный документ от нашей подруги Алисы, которой нравится помогать с примерами. Она бы использовала ваш открытый ключ для того, чтобы зашифровать его, и вы можете расшифровать его сразу (вам даже не нужен ее ключ).

```
$ gpg --decrypt document
```

Но если она документ еще и подписала, то вам понадобится ее ключ, чтобы подтвердить подлинность документа, поскольку *gpg* сообщит вам:

```
gpg: Can't check signature: No public key [Не могу проверить подлинность: нет открытого ключа]
```

Впрочем, он удостоверит подпись при наличии ключа на вашей связке ключей даже без вашего удостоверения, но и об этом вас предупредит:

```
gpg: Good signature from "Alice <alice@example.org>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
[Этот ключ не удостоверен доверенной подписью!]
gpg: There is no indication that the signature belongs to the owner
[Нет указания на то, что эта подпись принадлежит владельцу]
```

Представленное в скобках `"unknown [неизвестно]"` является значением, которое *GnuPG* рассчитал для ключа Алисы. Оно также выводится, если вы выводите свою копию ее ключа из вашей связки ключей:

```
$ gpg --list-key alice
pub rsa2048 2017-01-16 [SC] [expires: 2019-01-16]
UID [unknown] Alice <alice@example.org>
sub rsa2048 2017-01-16 [E] [expires: 2019-01-16]
```

Рассчитанное значение может быть `"unknown [неизвестное]"`, `"full [полное]"` или, для вашего собственного ключа, `"ultimate [крайнее]"`. Вы должны удостоверить ключ Алисы, используя любой подход, который сочтете приемлемым, а затем удостоверить его подлинность подписью:

```
$ gpg --edit-key alice
gpg> sign
```

Выйдите из редактора и сохраните изменения, когда вам предложат это сделать. Если вы снова укажете ключ, то увидите, что его рассчитанное значение теперь будет `"full"` —

```
UID [ full ] Alice <alice@example.org>
```

и оно создаст хороший отчет о подписи:

```
$ gpg --decrypt document
gpg: Good signature from "Alice <alice@example.org>" [full]
```

Вы также можете записать внутри подписи удостоверения, насколько тщательно вы удостоверили личность владельца. Если вы добавите в команду *gpg* опцию `--ask-cert-level`, процесс подписи

потребуется, чтобы вы выбрали уровень достоверности из следующего перечня:

- ▶ Я не буду отвечать.
- ▶ Вообще не проверять.
- ▶ Обычная проверка.
- ▶ Очень тщательная проверка.

Стандарт OpenPGP именовал эти уровни достоверности `"Generic [Общий]"`, `"Persona [Личный]"`, `"Casual [Обычный]"` и `"Positive [Положительный]"`, и они определяются как уровни от 0 до 3 по шкале подписи *GnuPG*. Вот пример тщательного удостоверения ("`3`"):

```
$ gpg --list-signs alice
UID [ full ] Alice <alice@example.org>
sig 3 814EE2DB21D58552 ...
```

Интерпретация уровней достоверности — дело субъективное, но спецификация OpenPGP (RFC4880, раздел 5.2.1) описывает их так:

- ▶ Общая достоверность применяется, когда уровень подтверждения, при наличии такового, неизвестен или не указан.
- ▶ Достоверность `Persona` — при отсутствии удостоверения.
- ▶ Обычная достоверность применяется, если сделано обычное подтверждение.
- ▶ Положительная достоверность применяется при получении существенного подтверждения личности.

GnuPG применяет общий уровень достоверности 0, если не потребовать иного (`--ask-cert-level`), за исключением самоуверения (подписи собственного ключа): тогда выдается положительный уровень достоверности 3. Обычно удостоверение `persona` при расчете значения достоверности ключа игнорируется, но вы можете использовать опцию `--mincert-level`, если решите это изменить. После удостоверения чье-то ключа обычно экспортируется и отправляется копия (`--armor` экспортирует печатные символы, в кодировке `base-64`, которые легко отправлять электронной почтой):



Вместо `"gpg --edit-key"`, за которым следует `"sign"`, можно использовать сокращение `"gpg --sign-key"`.

```
Terminal
created: 2017-01-18 expires: never usage: SCEA
trust: unknown validity: unknown
Primary key fingerprint: C27C 4025 FB62 F124 7B35 0177 4E4A 9E81 117A 600D

alice <alice@example.net>

Please decide how far you trust this user to correctly verify other users' keys
(by looking at passports, checking fingerprints from different sources, etc.)

 1 = I trust marginally
 2 = I trust fully

Your selection? 2

Please enter the depth of this trust signature.
A depth greater than 1 allows the key you are signing to make
trust signatures on your behalf.

Your selection? 2

Please enter a domain to restrict this signature, or enter for none.

Your selection? example.co.uk
```

▶ Доверенные подписи позволяют удостоверять и доверять. Но будьте готовы ответить на ряд дополнительных вопросов.

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



После импорта собственного ключа вам, возможно, понадобится явно указать доверие ему. Используйте `“gpg --edit-key”` для настройки доверия владельцу своего же ключа на `“ultimate”`.

```
$ gpg --armor --export alice | mail alice@example.org
```

Получатель может по своему усмотрению загрузить эту копию на сервер открытых ключей. В таком случае любой, кто использует этот ключ, сможет также увидеть ваше удостоверение и решить, полагаться ли на него, не занимаясь самостоятельным подтверждением подлинности ключа. Помните, что загрузка чужого ключа на сервер считается грубостью — следует отправлять удостоверенные вами ключи их владельцам и предоставить им самим решать, принимать ли ваши удостоверения публично.

Кому доверять?

Ключ-то вы удостоверить можете, но наш пример иллюстрирует, что это *GnuPG*, а не вы, принимает решение о подлинности ключа. *GnuPG* делает это, полагая, что вы уверены: кто бы его ни удостоверил, его подлинность подтверждена приемлемым для вас способом. Именно таково значение доверия в мире PGP — вы доверяете чьему-то подходу к подтверждению личностей других; это не обязательно должно означать, что вы бы доверили им свои сбережения!

Пока что у нас есть только доверие к собственному ключу, и помимо него единственными действительными ключами на нашей связке ключей будут те, которые мы удостоверяем. Но мы можем расширить доверие на те ключи на нашей связке, чьи владельцы тоже выполняют удостоверение приемлемым способом, и это разрешит *GnuPG* подтверждать ключи, удостоверенные ими. Как и ваше отношение к подтверждению, подобное доверие тоже субъективно и персонально; именно вы решаете, каково ваше доверие — вот несколько примеров:

- Я доверяю моему собственному ключу.
- Я доверяю вашему ключу, как своему.
- Я доверяю вашему ключу, но не самому по себе.
- Я не доверяю вашему ключу.

Вы можете передавать свое доверие двумя способами: `“owner trust [доверие к владельцу]”` — первичная классическая модель доверия [Classic Trust Model], или через более новую модель доверия PGP `“trust signatures [подписи доверия]”`. Вы в явной форме приписываете владельцу доверие к ключу отдельно от любого удостоверения, и оно хранится в вашей частной базе данных

доверия, тогда как подписи доверия являются удостоверениями, гарантирующими доверие в добавление к подлинности. Являясь удостоверениями, они хранятся на вашей связке ключей и активируются в тех случаях, когда ключ экспортируется и публикуется на серверах ключей; следовательно, подписи доверия могут стать публичными.

Приведенные примеры известны как уровни доверия; они могут быть `“ultimate [высший]”`, `“full [полный]”`, `“marginal [ограниченный]”`, `“never [никогда]”`, и, если вы не определили уровень, `“unknown [неизвестно]”`. Если вы чувствуете, что не можете доверять ключу, но всё же не хотите отказаться от него, есть еще один уровень, который можно использовать — `“undefined [неопределенно]”`.

Режим редактирования ключей инструмента командной строки *GnuPG* также используется для определения доверия владельцу. Если у вас на связке ключей есть ключ Алисы, и вы хотите ему доверять —

```
$ gpg --edit-key alice
gpg> trust
```

- Затем *GnuPG* предлагает пять опций для выбора доверия:
- `“I don't know or won't say”` -> `“undefined”` [Не знаю или не скажу -> неопределенно]
 - `“I do NOT trust”` -> `“never”` [Я НЕ доверяю -> никогда]
 - `“I trust marginally”` -> `“marginal”` [доверяю ограниченно -> ограниченно]
 - `“I trust fully”` -> `“full”` [полностью доверяю -> полное]
 - `“I trust ultimately”` -> `“ultimate”` [безоговорочно доверяю -> высшее]

Если вы выбрали четвертую опцию, присвоив полное доверие, то когда Алиса удостоверит ключ Блейка, это подтвердится и для вас:

```
$ gpg --list-signs blake
UID [ full ] Blake blake@example.org
```

Вы считаете Алису «рекомендателем» ключа Блейка, потому что она его удостоверила. Доверяя Алисе удостоверять этот ключ, вы также считаете ее «доверенным рекомендателем».

Время вопросов...

Используйте `tsign` вместо `sign`, если хотите сделать подпись доверия. *GnuPG* расширит процесс подписи несколькими дополнительными вопросами по доверию:

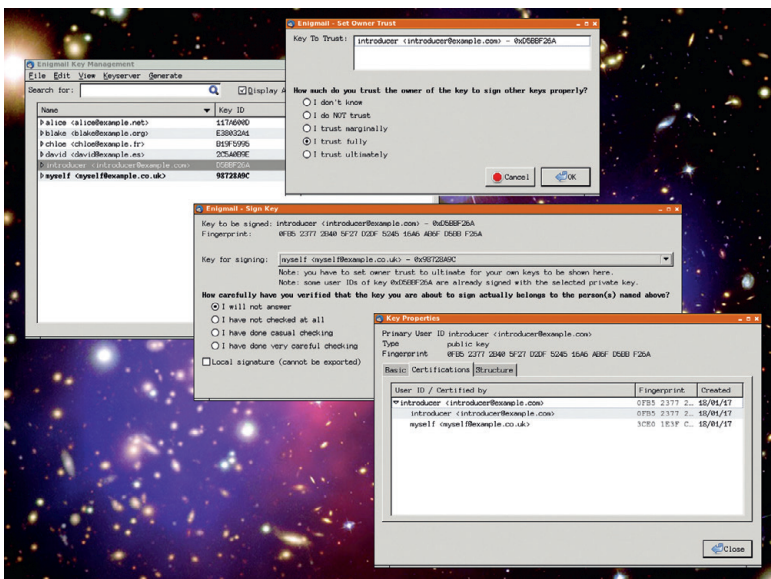
```
$ gpg --edit-key alice
gpg> tsign
```

Первый такой вопрос касается того, как вы доверяете владельцу ключа удостоверять другие ключи таким же способом, как и ранее описанный вопрос владельцу доверия, только ваши опции ограничены `“marginal”` или `“full”`.

Второй вопрос — о `“Delegated Trust [Делегированном Доверии]”`, которое позволяет вам доверенно подписывать ключ [trust-sign], чтобы любым ключам, так подписанным, тоже доверяли (например, я доверяю вам и тем, кому доверяете вы, но не тем, кому доверяют они). Подобная подпись доверия имеет два уровня делегированного доверия: доверие ключу, который вы подписываете, плюс одно дальнейшее соединение. Подписи доверия можно придать до пяти уровней делегированного доверия, где один уровень делегированного доверия равен доверию владельцу.

Как и с доверием владельцу, ключи, подписанные подписями доверия с одним уровнем делегированного доверия, являются доверенными рекомендателями. Ключи, подписанные двумя уровнями делегированного доверия, именуются `“Meta Introducers [Мета-рекомендателями]”`, а с тремя уровнями — `“Meta-meta Introducers [Мета-мета-рекомендателями]”`.

Третий и последний вопрос предлагает вам ввести домен для ограничения подписи. Вы можете оставить домен пустым (без



➤ Вам не нужна командная строка, чтобы удостоверить и приписывать доверие ключам. Инструменты GUI предлагают отличную работу с мышью, как в плагине *Enigmail* для *Thunderbird*.

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

Это ключ или сертификат?

Сертификат OpenPGP содержит открытый ключ, одну или более личностей пользователей и один или более открытых подключей. Однако сертификаты часто, хотя и ошибочно, считают ключами, как серверы ключей, которые на самом деле

обслуживают сертификаты. Правильные термины описаны в RFC4880 и в книге по PGP *Введение в криптографию [Introduction to Cryptography]*, которую можно найти на <http://cisweb.bristolcc.edu/~ik/Download/CIT18/IntroToCrypto.pdf>. Это неправильное

представление, кроме того, обсуждается на <http://bit.ly/2nzn7RD>.

В данном руководстве мы придерживались общепринятого языка и использовали в тексте слово «ключ».

ограничения) или ввести домен (например, example.com), которым ограничится делегированное доверие.

Еще одно незначительное различие между доверием владельцу и подписью доверия является то, что вы удостоверяете UID, тогда как доверие владельцу применяется к ключу. Доверие владельцу можно применить к ключу, имеющему подпись доверия, но только для повышения его уровня доверия.

Как бы вы ни приписали доверие (вы можете использовать доверие владельцу, подпись доверия или и то, и другое), GnuPG использует его для определения подлинности ключа и сохраняет его в вашей базе данных доверия. Ключ считается Полностью Подлинным, если он удостоверен хотя бы одним ключом с высшим или полным доверием, или тремя ключами с ограниченным доверием. Это стандартные параметры. Но вы можете их изменить, добавив записи в ваш файл настройки *GnuPG*.

GnuPG автоматически поддерживает вашу базу данных доверия, но вы можете потребовать немедленного обновления, и вот хороший способ увидеть ее обзор:

```
$ gpg --update-trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 1 trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: depth: 1 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 1f, 0u
gpg: next trustdb check due at 2019-01-16
```

Первая строка подтверждает параметры, описанные в предыдущих пунктах, и то, что используется модель доверия PGP. Следующие строки обобщают вашу сеть доверия. «depth [глубина]» представляет степени удаленности от вас; есть один ключ на глубине 0 — ваш собственный — который является подлинным, подписанным и имеет высшее доверие. Значения доверия — это счетчик ключей с неизвестным (q), отсутствующим (n), ограниченным (m), полным (f) или высшим (u) доверием.

Доверять ли Сети Доверия?

Одно из определяющих свойств Сети Доверия заключается в том, что ключи и их удостоверения публикуются, обычно через их загрузку на серверы открытых ключей. Ключи содержат личные данные (имена, адреса электронной почты, фото и все, что написано в поле комментариев), которые вам, возможно, не захочется выставлять на открытый сервер. Кроме того, удостоверения на вашем ключе раскрывают информацию о вашем социальном профиле — ваших знакомых, интересы и связи. А если вы включите уровни удостоверения при подписи ключа, то ваш подход к удостоверению тоже будет открыт для тех, кто собирает информацию с серверов ключей.

У серверов ключей также имеется важное свойство, которое нельзя упускать из вида: ключи с них нельзя удалить! Пускай вы и отозвали ключи или удостоверения, но они всё равно остаются нестираемым следом. Поэтому вам стоит как следует подумать, прежде чем загружать свой ключ на сервер ключей; это также объясняет, почему абсолютно неправильно без спроса загружать чей-то чужой ключ.

Важная часть процесса доверия также состоит в том, чтобы защитить себя, и вам стоит рассмотреть и положительные, и отрицательные аспекты сети доверия, чтобы здраво оценить их преимущества для вашей работы и цену в плане конфиденциальности.

Даже решив не использовать серверы ключей, вы всё равно можете иметь частную сеть доверия, распределяя ключи между своими знакомыми и удостоверяя их должным образом. Если вы хотите удостоверить ключ для личного пользования, никогда его не экспортируйте, можете применить вариант команды подписания ключей `lsign` (“local sign [локальная подпись]”), который мы употребляли в наших примерах. Он работает точно так же, но не позволяет случайно экспортировать удостоверения, которые лучше хранить частным порядком.

Но, с другой стороны, если вы желаете строить свою сеть доверия публично, загрузите свой ключ на сервер ключей, например, <https://sks-keyservers.net>. Серверы ключей синхронизируются с другими, способствуя распространению ключей. Вы, возможно, также решите заявить о своем желании подписывать ключи других (и чтобы они удостоверяли ваши) с помощью сайтов вроде <http://www.biglumber.com>.

В общем...

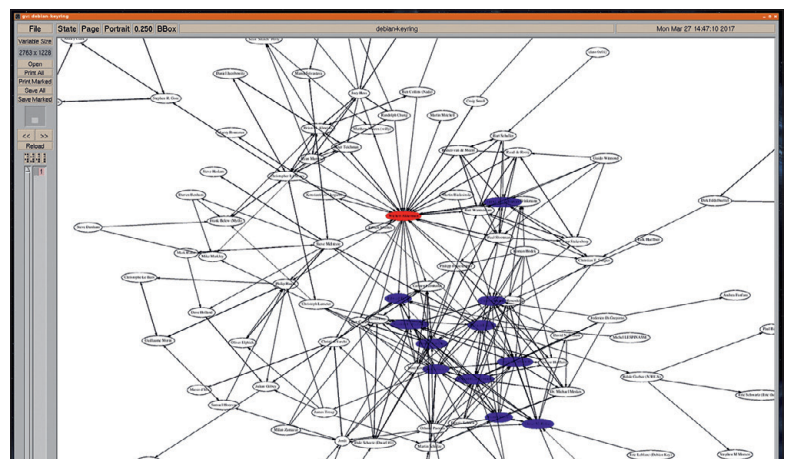
На те случаи, когда вы не уверены, что ключ подтвержден или проверен...

- ▶ У «владельца» ключа OpenPGP в нем указана его личная информация; имя, фото или электронная почта.
- ▶ Ключ «подлинный», если он действительно принадлежит указанному владельцу.
- ▶ Вы «подтверждаете» ключ, чтобы доказать его подлинность.
- ▶ Вы «подписываете» ключ, «удостоверяя», что вы его подтвердили.
- ▶ Вы называете подпись ключа «удостоверением», чтобы отличить ее от подписи на документе, сделанной с помощью ключа.
- ▶ Вы можете «доверять» владельцу ключа и принимать его удостоверение — или от него самого («полное»), или сделанное с помощью других («ограниченное» доверие).
- ▶ Ключ, имеющий удостоверения с достаточным доверием (один с полным или три с ограниченным), является «действительным».
- ▶ Вы можете «делегировать» решения о доверии владельцу действительного ключа, которому доверяете. **LXF**

Скорая помощь

База данных доверия обновляется автоматически; задав настройки доверия владельцу, вы можете безопасно выйти — “quit” — из “gpg --edit-key”.

Через месяц:
Защитите
свой IoT



▶ Вы можете визуализировать сеть доверия с помощью утилиты `dot` и скрипта “sig2dot” (см. <http://www.chaosreigns.com/code/sig2dot>). Данный пример показывает часть сети доверия связки ключей Debian.

Сервер LXF. На- правим трафик

Афнан Рехман погружается в мир сетей, чтобы выяснить, как превратить дистрибутив Linux в полноценно функционирующий роутер.



Наш эксперт

Афнан Рехман — студент, энтузиаст Linux и компьютерный умник, который с ходу всё ломает, так что вам этого делать не придется.



```
router@router-ZBOX: ~
GNU nano 2.5.3
# interfaces(5) file used by ifup(8) and ifdown
auto lo
iface lo inet loopback

# The WAN interface, above the USB port
auto enp3s0
iface enp3s0 inet dhcp

# The LAN interface, above the HDMI port
auto enp2s0
iface enp2s0 inet static
address 192.168.97.1
netmask 255.255.255.0
```

» В файле интерфейса вы увидите уже готовые строки. Скорее всего, они касаются петлевого интерфейса, и трогать их не стоит. Добавьте свои изменения под ними.

Наш урок послужит базовой инструкцией по настройке функционирующего роутера и предоставит вам платформу, которую вы сможете расширять по своему усмотрению.

Сперва обсудим главные компоненты любого роутера, предназначенного для упрощения домашней сети. Современный роутер часто представляет собой сочетание многих разных компонентов, собранных вместе, чтобы предоставить готовое решение. Роутеры, предлагаемые в магазинах, обычно состоят из собственно оборудования для маршрутизации, сетевого коммутатора (сетевые порты сзади) и беспроводной точки доступа, позволяющей беспроводному сигналу соединить все ваши беспроводные устройства. Нередко эти потребительские роутеры используют лишь сугубо необходимое оборудование и имеют ограниченные возможности для хранения, ОЗУ и мощность обработки. Эти мелкие компромиссы могут создать узкие места в вашей сети, особенно при работе с более высокими скоростями от вашего интернет-провайдера, например, 100 Мбит/с или выше. Роутеры с большей производительностью часто имеют заградительную цену. Решение нашего урока содержит только основные компоненты роутера, без коммутатора или беспроводной точки доступа. Однако вы можете добавить их отдельно.

Мал, да удал

Прежде чем обратиться к подробностям настройки, уделим минутку разговору об оборудовании. Некоторые из вас, наверное, желают знать, какое именно оборудование необходимо для создания функционирующего роутера. Некоторые из вас также, возможно, дивятся, кто в здравом уме будет использовать полноценный ПК как роутер. Если вас не очень привлекает идея поместить старое оборудование посреди своего кабинета — не волнуйтесь. Чудо современной технологии в том, что она становится всё меньше

Настало время взяться за вашу сеть. Мы находимся в неблагоприятной ситуации, поскольку любим технологии, но они часто нас подводят; и поэтому мы перебрали изрядное количество потребительских роутеров. В поисках замены на этот год нас осенила идея: а не ли создать собственный роутер на Linux, с полным контролем над функциями и настройками? Какая свежая идея! Мы тут же извлекли ПК из подвалов LXF и приступили к работе.

Идея создания роутера не столь нова, но ее популярность среди технарей растет, поскольку это — возможность выжать максимум производительности из настройки вашей маршрутизации, а также сохранить полный контроль в эпоху урезанных и контролируемых приложениями потребительских продуктов. Причина создания вашего домашнего решения в том, что оно делает обычную домашнюю или небольшую офисную сеть полностью вашей. Вы управляете всеми аспектами функциональности — от маршрутизации до IPtables и до сервисов NAT и DHCP. Вы даже можете добавить функции в роутер, чтобы управлять определенными типами трафика, скоростью и приоритетами устройств.

Скорая помощь

Возможно, вы заметите подключение, отдельное от ваших проводных подключений. Если в вашу машину встроена беспроводная карта или функция беспроводного доступа, обязательно их отключите.

Поиск неисправностей интернет-соединения

Впервые перезагрузившись после редактирования файла ваших сетевых интерфейсов, вы можете обнаружить, что ваше интернет-соединение заглохло и вам не удается его активировать. Вы можете заметить следующее сообщение про сетевой интерфейс: “device not managed [устройство неуправляемо]”. Дело здесь в том, что вы изменили конфигурацию сети в `/etc/network/interfaces` для двух ваших портов Ethernet. Это лечится редактированием файла

настройки сетевого менеджера. Можно открыть файл `/etc/NetworkManager/NetworkManager.conf` и изменить строку `managed=false`, поставив `true` вместо `false`. Сохраните файл и перезапустите сетевой менеджер командой

```
$ sudo service network-manager restart
```

В порядке альтернативы, советую установить здесь беспроводное соединение или выполнить все зачатки заранее. Если вам потребуется отказаться

от этого, просто откатите изменения в файле интерфейсов и перезагрузите компьютер. Если у вас приключились другие беды, обязательно загляните в официальную документацию Ubuntu на <https://help.ubuntu.com/community/IptablesHowTo>. Там дается масса ценной информации не только про устранение проблем, но и про редактирование конфигурации для добавления функциональности — например, ведения журналов.

и компактнее. Это применимо и к персональному компьютеру. Для нашего проекта мы использовали полноразмерный ПК, чтобы проверить идею, и затем перенесли всё на мини-ПК с дуальной сетевой картой (NIC) на Gigabit — примерно того же размера, что и портативный роутер для реальной работы.

Оборудование, использованное для сборки, включало валявшийся поблизости ПК с CPU Intel Celeron N3150 с 4 ГБ DDR3 O3Y и 64 ГБ SSD. Не перебор ли это? Естественно, перебор. Является ли это самой дешевой системой, которую можно найти для подобной настройки? Вероятно, нет. Вы определенно можете урезать расходы, разместив операционную систему на меньшем SSD, вращающемся жестком диске или даже SD-карте, и уж точно можно урезать объем ОЗУ. Процессор тоже может быть медленнее; всё зависит от ваших нужд. У нас эти компоненты просто были под рукой, и, честно говоря, мы хотели еще и добиться высшей производительности.

Самое важное — наличие у вас по крайней мере двух портов Ethernet, желательно со скоростью Gigabit. Причина проста: вам нужен один порт для соединения WAN (входящего из Интернета) и один для LAN (исходящего трафика в локальную сеть). Порт LAN можно подключить к коммутатору, чтобы упростить использование нескольких проводных устройств.

Теперь — об операционной системе. Мы используем Ubuntu Desktop, чтобы продемонстрировать нашу концепцию простейшим образом, и большая часть работы будет выполняться в командной строке. Linux в целом создан с учетом маршрутизации, что делает его выбор естественным. Как таковые, приведенные здесь инструкции можно адаптировать под практически любой обычный дистрибутив Linux. На слабых системах будет разумнее использовать минимальную установку, например, базовую Ubuntu Server или CentOS Minimal, чтобы снизить издержки от операционной системы, сохраняя всю мощь для собственно маршрутизации.

Настройка

Первый шаг, естественно, установить Ubuntu или выбранный вами дистрибутив. Это довольно просто, и онлайн есть множество руководств. Что бы вы ни решили использовать, мы рекомендуем обязательно выбрать долгосрочную поддержку, например, версию

```
router@router-ZBOX: ~
GNU nano 2.5.3 File: /etc/
# !/bin/sh
/sbin/iptables-restore < /etc/network/iptables
```

» Скрипт находится в новом файле и имеет всего две строки. Этот скрипт просто обновляет интерфейс, экономя наше время при перезапуске системы.

```
router@router-ZBOX: ~
File: /etc/network/iptables
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]

# enp3s0 is WAN interface and enp2s0 is LAN interface
-A POSTROUTING -o enp3s0 -j MASQUERADE

# NAT pinhole: HTTP from WAN to LAN
-A PREROUTING -p tcp -m tcp -i enp3s0 --dport 80 -j DNAT --to-destination 192.168.97.100:80

COMMIT

*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]

# Service rules
# UFW accept rules
-A INPUT -s 127.0.0.0/8 -d 127.0.0.0/8 -i lo -j ACCEPT
-A INPUT -m state --state ESTABLISHED -j ACCEPT

# enable traceroute reject
-A INPUT -p udp -m udp --dport 33434:33523 -j REJECT --reject-with icmp-port-unreachable

# DNS
-A INPUT -i enp2s0 -p tcp -dport 53 -j ACCEPT
-A INPUT -i enp3s0 -p udp --dport 53 -j ACCEPT
```

» Файл iptables запускается пустым, и мы добавим несколько строк. Обязательно добавьте комментарии, для ясности, если вам придется снова в него заходить.

Ubuntu LTS. Это обеспечит вас обновлениями безопасности на обозримое будущее, что очень важно для роутера, которым вы, возможно, будете пользоваться не один год.

Первое, что вам надо сделать после входа в систему — выяснить, где какой сетевой интерфейс. Возможно, придется взять ручку и бумагу. Экран должен показать несколько сетевых соединений, и одно из них отмечено как “lo”, как петлевое; о нем нам волноваться незачем. Наши соединения отмечены как “enp2s0” и “enp3s0”, и оба являются Ethernet-соединениями Gigabit. Ваше оборудование может быть другим, и имя интерфейса может тоже отличаться от того, что было у меня. Не забудьте записать эти имена, потому что вы будете их использовать по мере следования данному уроку.

Следующий шаг — настроить ваш сетевой интерфейс, когда вы выяснили, где какой. Введите следующую команду в свою консоль, чтобы открыть редактор:

```
$ sudo nano /etc/network/interfaces
```

Вас приветствует файл настройки, где уже есть несколько строк относительно петлевого интерфейса. Оставьте эти строки в покое и введите под ними следующее:

```
# Интерфейс WAN, над портом USB
auto enp3s0
iface enp3s0 inet dhcp

# Интерфейс LAN, над портом HDMI
auto enp2s0
iface enp2s0 inet static
address 192.168.97.1
netmask 255.255.255.0
```

Как видите, мы настроили и наш порт WAN (входящий), и наш порт LAN (исходящий). Я также поместил их комментариями, чтобы

Скорая помощь

Если здесь у вас появилась ошибка, проверьте, активирован ли у вас пользователь root в Ubuntu, поскольку часто это решает проблему.

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

знать, где какой. Это будет очень полезно позднее, когда мы будем употреблять эти интерфейсы для написания наших собственных правил маршрутизации. Порт LAN настраивается статическим IP-адресом, который должен соответствовать одному из ваших имеющихся на данный момент роутеров. Сетевую маску тоже надо определить, глядя на настройки вашего используемого роутера. Обе настройки могут отличаться от приведенных выше, в зависимости от вашей сети, так что обязательно проверьте всё хорошенько. Интерфейс WAN настраивается с помощью DHCP от вашего интернет-провайдера, поэтому мы просто написали строку выше и оставили ее как есть. Закончив, сохраните файл и перезагрузитесь.

Далее вам надо будет отредактировать файл `/etc/sysctl.conf` и раскомментировать (удалив символ `#`) строку `net.ipv4.ip_forward=1`.

Это разрешит переадресацию (проброс) пакетов для всех сетевых интерфейсов, что важно для переадресации пакетов между вашими сетями WAN и LAN. Сохраните это изменение и запустите `sudo sysctl -r`, чтобы обновить настройку.

Время таблиц

Теперь мы подошли к самой сути нашего урока: мы настроим `iptables`! `Iptables` — наиболее широко используемый брандмауэр

Linux уже долгое время, и мы здесь применим его для сортировки и ограничения входящего и исходящего трафика, что крайне важно, если мы собираемся подключаться к Интернету и вообще к другому устройству.

Первое, о чем надо позаботиться — это настройка правил переадресации пакетов, которые будут применяться перед запуском сетевых интерфейсов, что обеспечит немедленную переадресацию пакетов в случае перезапуска роутера. Сначала мы установим `iptables-persistent` — это пакет, позволяющий правилам `iptables` сохраняться после перезагрузки. Для его установки запустите следующую команду:

```
$ sudo apt-get install -y iptables-persistent netfilter-persistent
```

По завершении настроим скрипт запуска, который будет велять операционной системе запускать набор правил `iptables` до того, как станут доступными сетевые интерфейсы, чтобы роутер никогда не мог выйти онлайн или обратиться к Интернету без защиты набором правил `iptables`. Создайте скрипт командой

```
$ sudo nano /etc/network/if-pre-up.d/iptables
```

Заполните файл скрипта такими двумя строками:

```
#!/bin/sh
/sbin/iptables-restore < /etc/network/iptables
```

Теперь сохраните файл и по очереди запустите следующие команды:

```
$ sudo chown root /etc/network/if-pre-up.d/iptables
```

```
$ sudo chmod 755 /etc/network/if-pre-up.d/iptables
```

Первая сообщает системе, что скрипт находится в собственности `root`, а вторая — что он предназначен для записи `root` и для чтения/исполнения всеми.

Теперь создадим `iptables`, создав файл в `/etc/network/iptables` с помощью вашего любимого редактора. Заполните его следующими строками:

```
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
#enp3s0 - это интерфейс WAN, а enp2s0 - интерфейс LAN
-A POSTROUTING -o enp3s0 -j MASQUERADE
COMMIT
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
# Правила обслуживания
-A INPUT -j DROP
# Правила переадресации
-A FORWARD -j DROP
COMMIT
```



Помните, что ваши имена интерфейсов и IP-адреса, скорее всего, будут сильно отличаться от наших, поэтому измените их соответственно для каждого компьютера и сетевой среды.

```
router@router-ZBOX: ~
GNU nano 2.5.3 File: /etc/sysctl.conf
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1

#####
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
#net.ipv6.conf.all.accept_redirects = 0
#_or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
# net.ipv4.conf.all.secure_redirects = 1
#
```

➤ Именно в файле `sysctl` мы раскомментируем строку, разрешающую переадресацию портов. Настроек там много, поэтому постарайтесь не ошибиться строкой.

Куда отправляться отсюда...

Получив отлаженный роутер, вы можете захотеть пойти еще дальше. Теперь ваши возможности безграничны. В создании собственного роутера интересно то, что теперь вы сами отвечаете и за свое оборудование, и за программы, и у вас будут свободные ресурсы для добавления таких функций, как программы блокировки рекламы, и всё это будет «в одной посуде». Если вам захочется добавить функциональность, в Интернете есть множество

руководств, которые вам помогут. Вы также можете добавить новое оборудование — например, карты расширения сети, дополнительные устройства хранения и ОЗУ, даже дополнительные операционные системы и программы, чтобы создать многофункциональную серверную систему. Вы можете приобрести точку беспроводного доступа и сетевые коммутаторы для подключения всех своих проводных и беспроводных устройств к роутеру и в итоге

к Интернету. Делайте всё, что вашей душе угодно. Мы также рекомендуем в течение всего времени использования вашего роутера предпринимать действия по обеспечению нормальной работы. Например, очистка роутера от пыли каждые полгода снизит шум и выделение тепла, что продлит жизнь вашего оборудования и позволит вам наслаждаться работой высокопроизводительного и отлично настроенного роутера долгие годы.

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

Итак, здесь мы создали базовый скелет, включающий категории "nat" и "filter", каждая из которых заканчивается словом "commit". Важной штукой, которую делает этот начальный набор правил, является активация NAT, или Network Address Translation [Трансляция Сетевого Адреса]. NAT обеспечивает трансляцию сетевого адреса между локальными адресами вашей локальной сети и адресами с другой стороны роутера. Таким образом роутер узнает, куда отправлять пакет данных, входящих со стороны, и отправляет их должному клиентскому устройству в локальной сети.

Пока что мы не готовы выйти онлайн. И нам еще надо убедиться, что роутер раздает IP-адреса клиентам так же, как потребительский роутер. Эта часть очень проста. Сначала установим пакет сервера DHCP:

```
$ sudo apt-get install isc-dhcp-server
```

Далее откройте файл настройки `/etc/dhcp/dhcpd.conf` и добавьте такой компонент, для установки параметров адреса роутера, диапазона IP-адресов клиентов и широковещательного адреса: subnet 192.168.97.0 netmask 255.255.255.0 {
 range 192.168.97.50 192.168.97.199;
 option routers 192.168.97.1;
 option domain-name-servers 192.168.97.1;
 option broadcast-address 192.168.97.255;
}

Естественно, вы должны изменить эти адреса согласно вашим сетевым данным. Для клиентских IP-адресов вы можете задать любые параметры на свое усмотрение, и их диапазон может быть и большим, и маленьким. Чтобы применить настройки, мы просто запустили следующую команду:

```
$ sudo /etc/init.d/isc-dhcp-server restart
```

Пока что нам не хватает локального DNS, однако получить его еще проще. Просто запустите следующую команду, никакой настройки не понадобится:

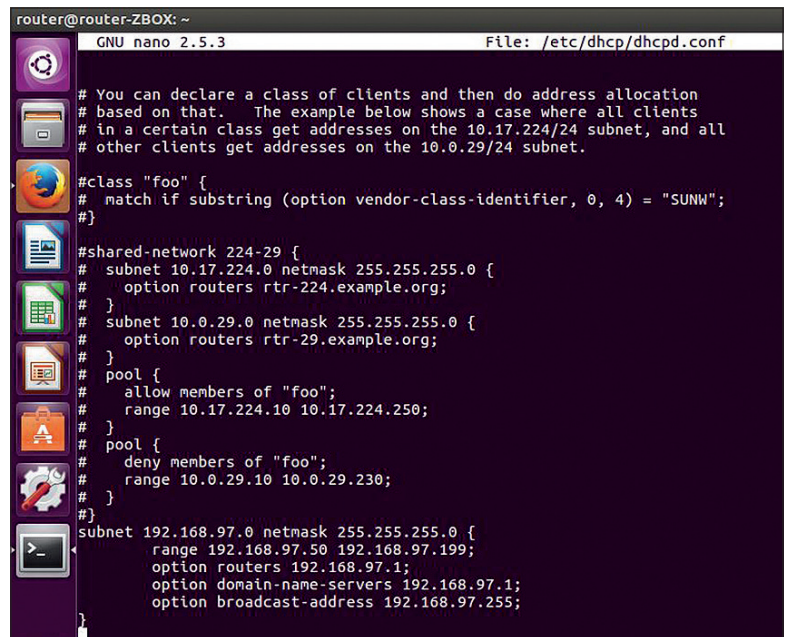
```
$ sudo apt-get install bind9
```

Послабления

Итак, всё в основном готово, и наш роутер уже способен справиться с обработкой запросов DNS, выдачей IP-адресов клиентам и переадресацией трафика. Однако наши правила пока настолько строгие, что он откажется всё это делать. Сейчас мы добавим несколько правил в наш набор, чтобы определить, какой трафик выходит в Интернет, что может идти в локальную сеть из Интернета, и каковы правила переадресации портов.

Итак, мы опять возвращаемся к редактированию `/etc/network/iptables` и начинаем с создания набора правил сервиса, правил переадресации и предварительной маршрутизации NAT. Наш полный набор правил показан ниже:

```
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
# еnp3s0 - это WAN, а еnp2s0 - это LAN
-A POSTROUTING -o еnp3s0 -j MASQUERADE
# Связь с NAT: HTTP от WAN в LAN
-A PREROUTING -p tcp -m tcp -i еnp3s0 --dport 80 -j DNAT --to-destination 192.168.97.50:80
COMMIT
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
# Правила сервиса
# Базовые правила по принятию
-A INPUT -s 127.0.0.0/8 -d 127.0.0.0/8 -i lo -j ACCEPT
-A INPUT -p icmp -j ACCEPT
```



В этом файле настройки DHCP уже немало информации. Добавьте ваши строки настройки в самом конце, да не трогайте ничего другого.

```
-A INPUT -m state --state ESTABLISHED -j ACCEPT
# Активация запрета traceroute
-A INPUT -p udp -m udp --dport 33434:33523 -j REJECT --reject-with icmp-port-unreachable
# DNS
-A INPUT -i еnp2s0 -p tcp --dport 53 -j ACCEPT
-A INPUT -i еnp2s0 -p udp --dport 53 -j ACCEPT
# SSH
-A INPUT -i еnp2s0 -p tcp --dport 22 -j ACCEPT
# Запросы DHCP-клиента - принимать от LAN
-A INPUT -i еnp2s0 -p udp --dport 67:68 -j ACCEPT
# Отбрасывать весь прочий входящий трафик
-A INPUT -j DROP
# Правила переадресации
# Проброс пакетов по соответствующим соединениям
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
# Проброс от LAN к WAN
-A FORWARD -i еnp2s0 -o еnp3s0 -j ACCEPT
# Разрешить трафик от нашего NAT
-A FORWARD -p tcp -d 192.168.97.100 --dport 80 -j ACCEPT
# Отбрасывать весь прочий переадресованный трафик
-A FORWARD -j DROP
COMMIT
```

Набор правил сервиса находится в области фильтра и будет создавать правила для того, что может принять роутер и что он может перенаправить в локальную сеть. Здесь мы также разрешаем доступ SSH, чтобы после настройки роутера мы могли удаленно войти и внести изменения, не подключая монитор и клавиатуру.

Раздел, озаглавленный «Правила переадресации», велит роутеру перенаправлять трафик в LAN и из LAN в WAN для исходящего трафика. Кроме того, мы добавим строку в этот раздел NAT для создания точки NAT, дающей роутеру указания перенаправлять весь произвольный трафик из Интернета на локальный компьютер по указанному адресу. Проверьте наличие правил PREROUTING и FORWARDING в должном месте.

Для завершения перезапустите `iptables`, скомандовав

```
$ sudo /etc/network/if-pre-up.d/iptables
```

Теперь у вас всё должно сработать! Наслаждайтесь новым и улучшенным взаимодействием с роутером! **LXF**

TruPax: файлы шифровать легко

Нейт Дрейк представляет универсальную утилиту Java, способную зашифровать ваши файлы за три простых шага в большой файл контейнера.



Наш эксперт

Нейт Дрейк — технический журналист-фрилансер, специализирующийся на кибер-безопасности и ретро-технологиях.



Скорая помощь
Прежде чем нажать Add Folder, убедитесь, что вы отметили окошко Include Subfolders, чтобы заодно скопировать в новый контейнер все директории внутри.

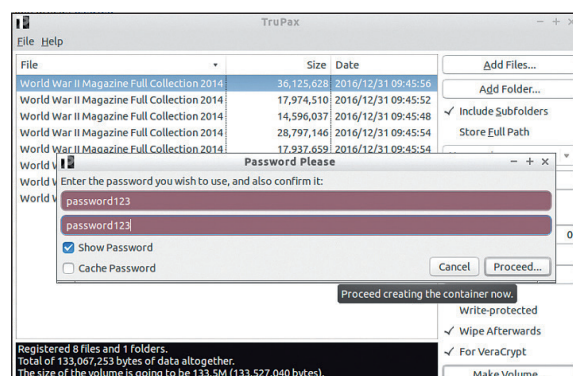
Для не помешанных на шифровании, *VeraCrypt* — это программа, позволяющая создавать зашифрованные контейнеры любого размера, внутри которых можно помещать ваши фото, банковские реквизиты или любые другие данные, которые не готов увидеть мир. *VeraCrypt* монтирует такие контейнеры с помощью пароля, а это означает, что только вы сможете получить доступ к вашим файлам. Для особо параноидальных людей есть даже опции комбинирования нескольких способов шифрования и использования файлов ключей в дополнение к паролю.

VeraCrypt требует от вас заранее продумать размер необходимых вам файлов. После создания зашифрованного контейнера его размер изменить невозможно. Более того, если вы создадите контейнер, будучи полностью уверенным, что у вас хватит места для всех ваших файлов, а потом обнаружите, что места недостаточно, вам не останется ничего другого, кроме как начать процесс создания тома заново.

В лучшем случае, это раздражает, и в особенности если вы решили сохранить свой зашифрованный контейнер в облачном сервисе вроде Dropbox, поскольку процесс загрузки придется перезапускать.

TruPax предлагает решение создать на вашем компьютере избыточно огромные блоки данных. Это приложение Java можно использовать для выбора файлов или папок перед созданием тома, чтобы создать блок должного размера. Есть даже возможность добавить немного свободного места, если вы считаете, что оно вам понадобится.

На нашем уроке мы выясним, как использовать *TruPax* для создания зашифрованного контейнера требуемого размера, а затем открыть его в *VeraCrypt*. Для продолжения вам понадобится



» Если мир не готов узнать вашу подноготную энтузиаста военной истории, создайте для своей коллекции журналов защищенный контейнер. Отметьте show password [показать пароль] для отображения того, что вы ввели.

установить на ваш компьютер Java (или его простую разновидность из OpenJDK от Oracle — см. ниже). После создания тома вам также понадобится установленный *VeraCrypt*, чтобы том смонтировать.

VeraCrypt можно скачать с <http://veracrypt.codeplex.com>. Сайт сам по себе предоставляет отличную документацию для начала, либо можете ознакомиться с предыдущим учебником в **LXF218**.

Этот учебник был написан для Ubuntu Linux, но Java и *VeraCrypt* совместимы со всеми версиями Linux, так что у вас не должно быть проблем с запуском *TruPax*.

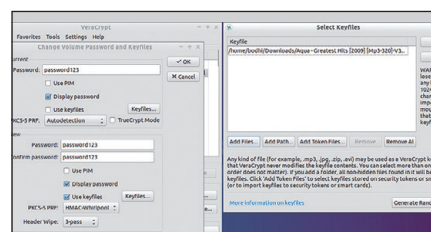
Другим большим преимуществом *TruPax* является то, что для простого создания тома или распаковки его содержимого

TruPax и не только

Удобство *TruPax* для создания томов правильного размера, увы, отчасти создается за счет безопасности. При желании вы можете добавить файл ключа. Тогда для монтирования вашего тома *TruPax* в *VeraCrypt* будет использован как ваш пароль, так и файл. Это — форма двухфакторной аутентификации (нечто имеющееся у вас и нечто вам известное), значительно снижающая шансы злоумышленника получить доступ к вашему тому. У *VeraCrypt* также есть генератор случайных файлов ключа. Перейдите в Tools > Keyfiles для его запуска. Можно использовать один или более файлов. Помните, что первые 64 КБ файла нельзя изменять,

иначе контейнер будет не открыт. Вот почему изображения и музыкальные файлы работают лучше. *VeraCrypt* не изменяет файл, и если вы, например, выберете в качестве файла ключа MP3 *Doctor Jones* от группы Aqua, вам не избежать изучения самого файла, чтобы выяснить, использовался ли он таким образом. В *VeraCrypt* выберите Volumes > Change Volume password, чтобы приступить к работе. Введите свой пароль в поле раздела Current, а затем снова в разделе New. Загляните в Use Keyfiles и нажмите на кнопку Keyfiles для выбора нужных файлов. Порядок выбора роли не играет. Перед нажатием OK вверху справа перейдите в выпадающее меню

PKCS-5 PRF в разделе New и выберите HMAC-Whirlpool, чтобы использовать хэш-сумму, созданную совершенно независимо от наших друзей в АНБ.





По умолчанию *TruPax* создает контейнеры, совместимые с предшественником *VeraCrypt* — *TrueCrypt*, у которого был ряд уязвимостей в системе безопасности. Так что при создании тома для надежности обязательно отметьте *For Veracrypt*.

куда-либо в систему не нужны привилегии администратора. А вот *VeraCrypt* потребует ваш пароль администратора для монтирования контейнера, чтобы вы могли редактировать свои файлы.

Одним из преимуществ *TruPax* перед *VeraCrypt* является безумная простота в использовании. После запуска просто нажмите *Add Files* или *Add Folder*, чтобы загрузить ваши данные в основное окно.

После добавления файлов вы увидите уведомление в нижней части окна, сообщающее размер будущего контейнера. Если вам нужно место для большего количества файлов, обратитесь к окну *Free Space* в правой части окна. Оно довольно интуитивно и располагает такие значения, как '500m' или '2g'.

Опционально вы можете дать тому метку. Перед нажатием на кнопку *Make Volume* внизу справа убедитесь, что вы отметили *Wipe Afterwards* [Очистить впоследствии], если хотите безопасно удалить файлы-оригиналы (см. ниже).

Вас попросят задать пароль для тома, а затем *TruPax* примется создавать ваш контейнер.

Очистка и извлечение

Если вы специально не попросите *TruPax* не делать этого, убрав флажок от окошка *Wipe Afterwards*, *TruPax* безопасно удалит файлы-оригиналы после их копирования в защищенный контейнер. Прежде чем начать, позаботьтесь, чтобы файлы копировались в надежное место.

Для создания незашифрованной копии ваших файлов можно использовать *File > Extract* в меню вверху слева, при условии, что вы знаете пароль. В принципе, если вам надо только зашифровать файлы для долгосрочного хранения, например, для резервных копий, может оказаться так, что вам вообще незачем обращаться к *VeraCrypt*.

Если, что более вероятно, вы захотите иметь возможность время от времени редактировать эти файлы и добавлять еще что-то в свой контейнер, вы можете использовать все тот же *TruPax*, чтобы изначально создать контейнер, а затем смонтировать его с помощью *VeraCrypt*.

TruPax написан на Java, и следовательно, требует установленной JRE (Java Runtime Environment). Руководство перечисляет, как добавить репозиторий для этого, а также как установить Java 8, что является минимумом для запуска *TruPax*.

Люди, озабоченные безопасностью и/или те, кто по-настоящему предан свободному и открытому ПО, могут предпочесть установить *OpenJDK*, тоже разработанный добрыми людьми из Oracle, но не содержащий закрытого кода. Если вы предпочитаете этот вариант, игнорируйте Шаг 1 данного руководства и запустите команды

```
sudo add-apt-repository ppa:openjdk-r/ppa
sudo apt-get update
sudo apt-get install openjdk-8-jre
```

Если вы — один из тех, кто бросается прямо к учебникам, не прочитав основную статью (вам виднее, кто вы), знайте, что Oracle Java Runtime Environment и *OpenJDK* могут вполне успешно уживаться на одном компьютере. Запустите команду `sudo update-alternatives --config java`, если хотите сменить программу Java по умолчанию на *OpenJDK*. Выберите требуемую цифру, нажмите *Enter*, а затем перезагрузите компьютер, чтобы изменения вступили в силу.

Tru лжет

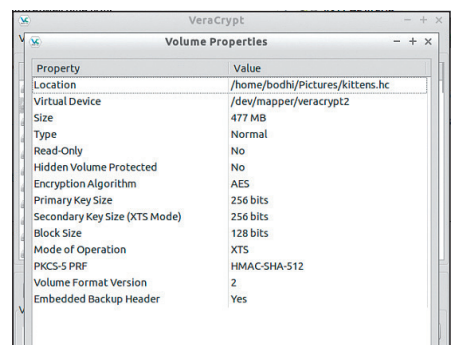
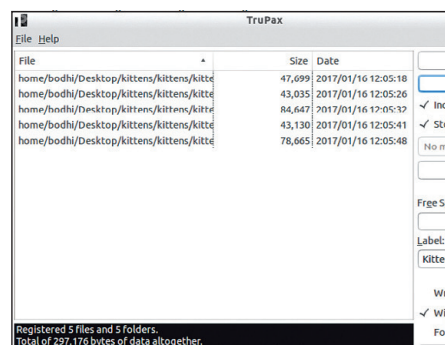
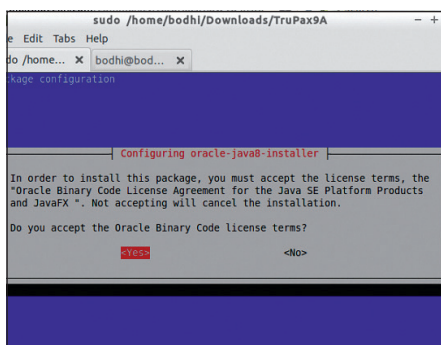
[обыгрывается созвучие части названия *TruPax* со словом true — «правда», — прим. пер.] Сам *TruPax* может только создавать зашифрованные контейнеры для файлов и папок, а затем целиком извлекать их куда угодно. Это не слишком удобно для повседневного использования, потому-то мы и рекомендуем использовать его вместе с *VeraCrypt*.

В отличие от *VeraCrypt*, *TruPax* не умеет создавать скрытые тома, при которых вы можете иметь два пароля для контейнера — один из них можно смело сообщить неприятелю, поскольку он ведет к безобидным данным, а другой ведет к вашим реальным файлам. При желании вы, конечно, можете использовать *VeraCrypt* для создания скрытого тома и скопировать туда свой контейнер *TruPax*.

Если вы последуете шагам из врезки *TruPax* и не только для увеличения безопасности вашего контейнера посредством добавления файлов ключа, вы не сможете извлекать содержимое с помощью *TruPax*.

Для более подробной информации про настройку *TruPax* обязательно прочитайте [trupax_EN.html](#), скачиваемый вместе с программой, или же загляните на сайт разработчика — [www.coders-lagoon.com](#). **LXF**

Шифрование с помощью TruPax



1 Скачайте JRE & TruPax

Откройте Terminal и добавьте PPA для Java командой `sudo add-apt-repository ppa:webupd8team/java`. Затем запустите `sudo apt-get update` и `sudo apt-get install default-jre oracle-java8-installer`. Перейдите на <http://coderslagoon.com> для скачивания *TruPax*. Щелкните правой кнопкой по файлу, выберите *Extract*, а затем используйте *cd* для перехода в его папку — например, `cd /home/bodhi/Downloads/TruPax9A`.

2 Запустите TruPax

Выполните `java -jar trupax.jar` для запуска *TruPax*. Выберите себе язык (предлагаются немецкий и английский). Нажмите *Add Files* или *Add Folder*, чтобы выбрать данные для шифрования. Когда будете готовы задать пароль, нажмите *Make Volume*. *TruPax* спросит вас, где сохранить файл, и попросит дать файлу имя, например, **kittens hc**. В качестве расширения можете использовать любые имена, которые взбредут вам в голову.

3 Смонтируйте в VeraCrypt

Запустите *VeraCrypt*, затем нажмите *Select File*. Перейдите в свой том *TruPax*, затем нажмите *Mount*. Введите пароль, заданный вами ранее. Узнайте больше о вашем новом томе, выбрав *Volumes > Volume Properties*. Это покажет, где смонтирован контейнер, например, **/dev/mapper/veracrypt2**. В этом пункте вы, возможно, захотите повысить безопасность (см. *TruPax* и не только).

ЧАСТЬ 7

IoT: Дом, который послушен нам

Разочарованный контролем через web-сервер, Максим Черепанов выбирает JavaScript как язык сценариев для браузера.



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».



Пора, пора браться за работу. Вероятно, где-то в спальне у вас уже живет Arduino с радиомодулем и подключенным ночником, а BananaPi/RaspberryPi прописался в лаборатории или гостиной — там, где вы изучаете Linux. Связь-то между ними есть, но я уверен, что контроль на web-сервере пока не работает так, как надо. Ведь в прошлый раз мы перенесли два канала с пинов командного контроллера на периферийный контроллер, а контроль исполнения не скорректировали. Займемся сегодня этим, и еще кое-чем другим.

Актуализация данных

Отвлечемся на время от конкретной задачи и немного порассуждаем об актуальности данных и контроле через web-сервер. Поскольку наш ПД проектируется как многопользовательская система, вполне вероятен случай, когда включить или выключить кондиционер захотят два или больше членов вашей семьи (у меня такое летом бывает постоянно). При открытии страницы `index.php` система считает данные о состоянии нагрузок и отобразит их картинками в виде красных или серых индикаторов. Если вы захотите изменить состояние и нажмете кнопку, то произойдет следующее: с `index.php` вас перенаправят на один из файлов `on.php` или `off.php`. Там после действий по управлению нагрузкой строка `header("Location: ".$_SERVER["HTTP_REFERER"]);` принудительно заставит web-сервер обратиться к заглавной странице. Поскольку `on.php/off.php` не выводят ничего на экран, то

просто произойдет перезагрузка страницы (про это мы говорили в 4-й части нашего цикла), и индикаторы актуализируются.

Допустим, что в момент времени, когда вы уже открыли web-лицо ПД, но еще не нажали кнопки, от другого пользователя придет команда по e-mail, или он дозвонится к ПД через Asterisk. Но web-сервер про изменение статуса нагрузок ничего не знает, и информация о части нагрузок будет неактуальна. В этом нет ничего криминального, если вы не управляете ничем важным. А если это гаражные ворота? Я решил не включать фантазию для моделирования ситуации, когда отсутствие контроля будет критичным. Просто это неправильно, и всё.

Что можно предложить как выход? Самое простое — заставить сервер перезагружать заглавную страницу через определенный период времени, допустим, через 5 секунд. Для этого достаточно между тегами `<head>` и `</head>` файла `index.php` добавить строку `<meta http-equiv="refresh" content="5">`

Работает превосходно, как будто кто-то невидимый нажимает каждые пять секунд клавишу F5. Но меня такое раздражает, хоть это и субъективно. А объективно: постоянный трафик с полной перезагрузкой страницы не способствует удобству в работе. Если вы в локальной сети, то это ни на что не влияет и практически незаметно, кроме возникающего раз в 5 секунд сообщения о передаче данных. А если вы в другом городе, да соединились по 3G (или по 2G)? И дорого, и медленно.

Альтернатива есть, и имя ей — JavaScript как язык сценариев для браузера. Я стараюсь по возможности сокращать использование этой технологии в своих проектах. Причин несколько; главная — это рефлекс от использования браузеров в Windows (как давно это было...): когда-то вредоносные скрипты были настоящей чумой для этой ОС. С тех пор у меня в Firefox всегда установлено дополнение NoScript, отключаемое только вручную. Разумеется, современные антивирусы умеют отслеживать на лету активное содержание браузеров и в случае опасности блокируют их. Главное — определить критерий опасности, и не перегнуть палку. Лично у меня были случаи, когда антивирус помещал в карантин совершенно легальные файлы активации программ.

Тем не менее, современный Web немыслим без активного содержимого сайтов — это данность, от которой никуда не уйти. Но на мой взгляд, там, где можно применить PHP, не стоит использовать JavaScript. Я думаю, все знают их коренное отличие? Первый работает на серверной стороне, и браузер просто отображает результат. Второй работает на клиентской стороне, т.е. web-сервер грузит на ваш компьютер активное содержимое, и оно автономно запускается внутри браузера. Страшно? А должно быть... Конечно, в каждой шутке есть доля шутки, но в свое время не одна система была «убита» подло загруженным скриптом.

Я надеюсь, что достаточно запугал тех, кто мало имел с этим делом. В любом случае, с JavaScript нужно быть осторожным.

АЖАХ — великий и ужасный

От преамбулы — к делу. Не будем в очередной раз изобретать велосипед, а воспользуемся знаниями, наработанными человечеством до нас. Знакомьтесь: АЖАХ — Asynchronous Javascript and XML — «асинхронный JavaScript и XML»! Это технология, использующая фоновый обмен данными браузера с web-сервером. В результате при обновлении данных web-страница не перезагружается полностью, а подгружаются только нужные нам участки. За счет этого уменьшается время отклика, и web-приложение по интерактивности больше напоминает настольное. Несмотря на то, что в названии технологии присутствует буква X (от слова XML), использовать XML вовсе не обязательно. Под АЖАХ подразумевают любое общение с сервером без перезагрузки страницы, организованное при помощи JavaScript.

Чтобы продолжить, нам необходимо узнать еще один новый термин: jQuery — JavaScript-framework, библиотека, позволяющая более удобно использовать некоторые возможности JavaScript, такие как: создание визуальных эффектов, обработка событий, работа с DOM и поддержка АЖАХ. Можете считать, что это просто очень большой набор полезных функций, которые позволяют удобно делать то, что обычно требуется делать при помощи JavaScript.

Сначала надо скачать саму библиотеку (желательно с сайта разработчиков <http://jquery.com/download/>), при необходимости разархивировать и перенести ее в ту же папку, где лежат наши web-страницы, т.е. в папку, где находится файл `index.php`. Это не обязательно, но удобно. Обратите внимание на то, что многие сайты предлагают скачать jQuery именно у них. Я далек от мысли, что внутри интегрируют вредоносное содержание или скрипты, перенаправляющие браузер на определенный сайт (хотя и это не исключено). Просто на сайте разработчика будет наиболее свежая версия.

Теперь нам надо подключить jQuery к HTML- или PHP-странице. Для этого существует тег `<script>`, который и отвечает за подключение файлов внешних скриптов:

```
<script type="text/javascript" src="jquery.js"></script>
```

Обязательно сверьте имя файла, оно может отличаться в зависимости от даты разработки, например, `jquery-3.2.1.min.js`. Разумеется, имя файла должно быть одним и тем же в строке подключения и в папке web-сервера.

Всё, с этого момента можно пользоваться данной библиотекой, не задумываясь, как именно осуществляются те или иные действия. Подробно о jQuery можно узнать на большом количестве сайтов и из книг. Нам же пока требуется только осуществлять загрузку данных на страницу, не перезагружая ее.

Для понимания того, как это будет происходить, давайте напишем простейший пример. Создаем файл в директории web-сервера: `# touch /var/www/test.php` (или `/var/www/html`, в зависимости от настроек Apache).

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
Charset=UTF-8">
<script type="text/javascript" src="jQuery.js"></script>
</head>
<body>
<div id="content"></div>
<script>
function show()
```

```
{
$.ajax({
url: "control",
cache: false,
success: function(logo){
$("#content").html("");
}
});
}
$(document).ready(function(){
show();
setInterval('show()',1000);
});
</script>
</body>
</html>
```

И создадим еще один файл — `#touch /var/www/control` — со всего лишь одной строкой содержания:

```
red.jpg
```

В директории должны были остаться 2 файла-индикатора («лампочки»), `red.jpg` и `grey.jpg`.

И откройте этот файл через браузер (IP — просто к примеру): `http://192.168.0.30/test.php`. Вы увидите изображение файла, указанного в файле `control`. А теперь поменяйте в файле строку на `grey.jpg`, не закрывая браузер. Изображение в браузере изменится, а страницу-то мы не обновляли!

Теперь кратко пробежимся по файлу-примеру:

■ В теле документа создается контейнер, в который мы будем загружать контент:

```
<div id="content"></div>
```

■ Функция `$(document).ready()` требуется для корректной работы jQuery, к тому же в ней мы можем выполнить все приготовления к работе программы. В нашем случае мы вызываем функцию `show()`, в которой прописан механизм получения контента из другого файла, и настраиваем таймер так, чтобы функция `show()` вызывалась один раз в секунду:

```
$(document).ready(function(){
show();
setInterval('show()',1000);
```

Функция `show()` состоит из обращения к функции `$.ajax()` с определенным рядом параметров, которая позволяет нам в фоновом режиме получить информацию из внешнего файла на сервере:

```
url: "control"
Обращается к файлу control для получения контента.
cache: false
Результаты запросов не кэшируются.
success: function(html){
$("#content").html("");
}
```

При успешном выполнении запроса управление переходит функции, которая получает контент в качестве параметра и записывает в контейнер картинку по тому имени, которое получено с сервера.

Рихтуем заглавную страницу

Очень надеюсь, что всё понятно объяснил, и теперь приведу сразу весь переделанный (не последний раз) файл `index.php`:

```
<html>
<head>
<title>Послушный дом тест с кнопками</title>
<meta charset="utf-8">
<script type="text/javascript" src="jQuery.js"></script>
```

»

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

```

<link rel="stylesheet" href="styles.css">
</head>
<body>
<p>Здравствуйте, я ваш Послушный дом</p>
<?php $channels=4;?> <!-- количество каналов -->
<table>
<?php
$A=array("кондиционер", "обогреватель",
"увлажнитель", "телевизор");
$pin=0;
while ($pin<$channels)
{
echo "<tr><th><form action='on.php' method='post'><p>
<button class='green' name='pin' value='$pin'>Включить
$A[$pin]</button>
</form></th><th>";
echo "<div id='content$pin'></div>";
echo "</th><th><form action='off.php' method='post'><p>
<button class='red' name='pin' value='$pin'>Выключить
$A[$pin]</button>
</form></th></tr>";
$pin++;
}
?>
</table>
<?php
echo "<script>
function show()
{";
$pin=0;
while ($pin<$channels) {
echo "
$.ajax({
url: \"control$pin\",
cache: false,
success: function(indicator){
$(\"#content$pin\").html(\"<img src='\"+indicator+\"'\");
}
});";
$pin++;}
echo "
$(document).ready(function(){
show();
setInterval('show()',1000);
});
</script>
?>
</body>
</html>

```

Создайте в директории web-сервера 4 файла: **control0**, **control1**, **control2** и **control3**, впишите в каждый для начала строку `grey.jpg`.

Откройте в браузере **index.php**; вы увидите знакомое лицо Послушного Дома. Только теперь при изменении любого из файлов **control0-3** будут меняться и индикаторы.

Получение данных для web-сервера

Теперь надо сделать так, чтобы в этих файлах своевременно менялась информация. Первое — это придется пересобрать `nrf24`, файл на C (тот, что обеспечивает обмен с Arduino). Меняем только одну строчку —

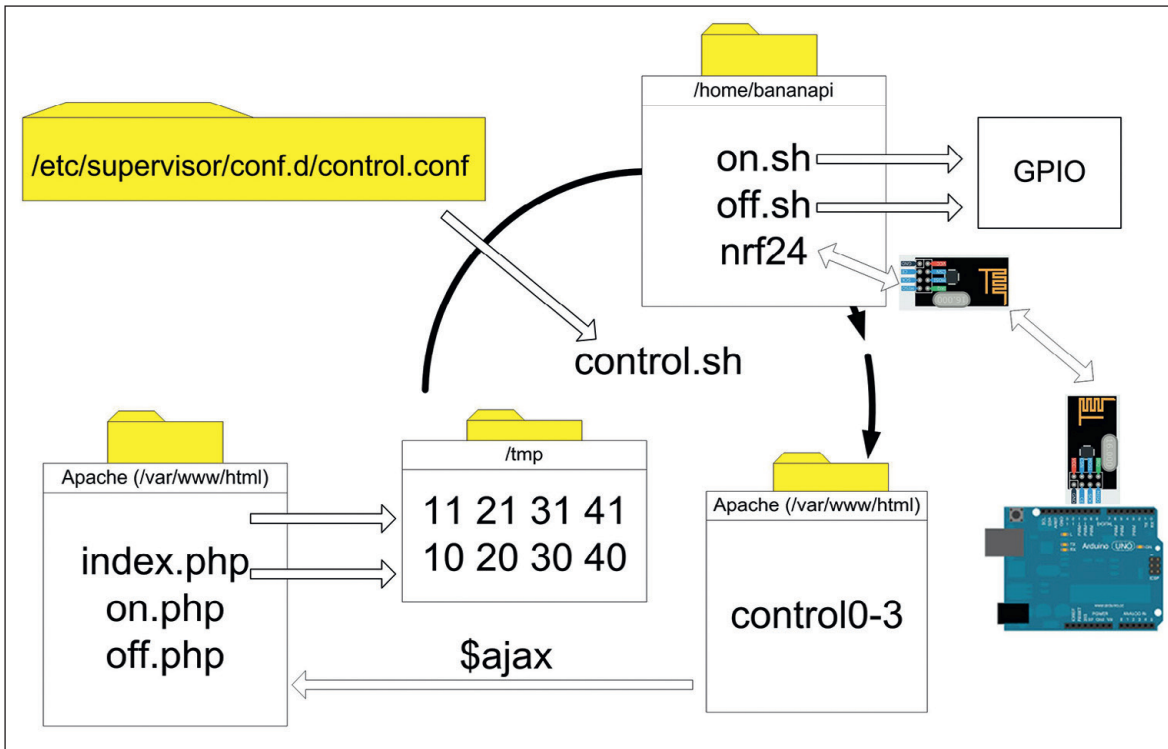
```
printf("Got response %lu\n\r", got_time);
```

на следующую:

```

printf("%lu", got_time);
И вспоминаем, что вывод "0" — оба канала не горят, "1" и "10" —
какой-либо один, "11" — оба.
После этого добавляем еще кое-что в скрипт control.sh в до-
машней директории (хотя он может находиться и в другом месте):
#!/bin/bash
while true
do
test -e /tmp/11 && CONTROL1=/home/bananapi/on.sh 0` && rm /
tmp/11
test -e /tmp/10 && CONTROL1=/home/bananapi/off.sh 0` && rm /
tmp/10
test -e /tmp/21 && CONTROL2=/home/bananapi/on.sh 2` && rm /
tmp/21
test -e /tmp/20 && CONTROL2=/home/bananapi/off.sh 2` && rm /
tmp/20
test -e /tmp/31 && CONTROL34=/home/bananapi/nrf24 31` && rm
/tmp/31
test -e /tmp/30 && CONTROL34=/home/bananapi/nrf24 30` &&
rm /tmp/30
test -e /tmp/41 && CONTROL34=/home/bananapi/nrf24 41` && rm
/tmp/41
test -e /tmp/40 && CONTROL34=/home/bananapi/nrf24 40` && rm
/tmp/40
case $CONTROL1 in
0)
echo "grey.jpg" > /var/www/control0
CONTROL1=99
;;
1)
echo "red.jpg" > /var/www/control0
CONTROL1=99
;;
esac
case $CONTROL2 in
0)
echo "grey.jpg" > /var/www/control1
CONTROL2=99
;;
1)
echo "red.jpg" > /var/www/control1
CONTROL2=99
;;
esac
case $CONTROL34 in
0)
echo "grey.jpg" > /var/www/control2
echo "grey.jpg" > /var/www/control3
CONTROL34=99
;;
1)
echo "red.jpg" > /var/www/control2
echo "grey.jpg" > /var/www/control3
CONTROL34=99
;;
10)
echo "grey.jpg" > /var/www/control2
echo "red.jpg" > /var/www/control3
CONTROL34=99
;;
11)
echo "red.jpg" > /var/www/control2
echo "red.jpg" > /var/www/control3
CONTROL34=99

```



► Рис. 1. Структурная схема передачи данных для веб-сервера.

```
;;
esac
sleep 1
done
```

Поясним: CONTROL1, CONTROL2, CONTROL34 — переменные состояния каналов, 99 им присваивается каждый раз при использовании состояния (после записи в файл), чтобы не происходило постоянное обращение к диску (карте). Данные в переменные приходят двойкой: из GPIO при прямом подключении и из модуля nrf24 при подключении нагрузки к Arduino. Всё остальное достаточно просто: в условиях case происходит сортировка результатов, и в зависимости от нее проводится запись в файлы-индикаторы. Не забудьте, что **control.sh** работает в режиме демона, и при изменении надо перезагружать супервизор:

```
# service supervisor restart
```

После этого всё должно заработать как надо.

Если вы всё же запутались, я нарисовал, как происходит обмен данными (не включены почтовый сервер, Asterisk и управление с помощью СМС) — см. рис. 1.

Ложка дегтя

Не могу не покаяться. По окончании этого этапа откройте веб-страницу ПД в Mozilla Firefox, затем нажмите Ctrl+Shift+K. Откроется консоль браузера, и вы увидите в ней один раз в секунду выпадающие ошибки с текстом об ошибках синтаксического анализа XML (рис. 2).

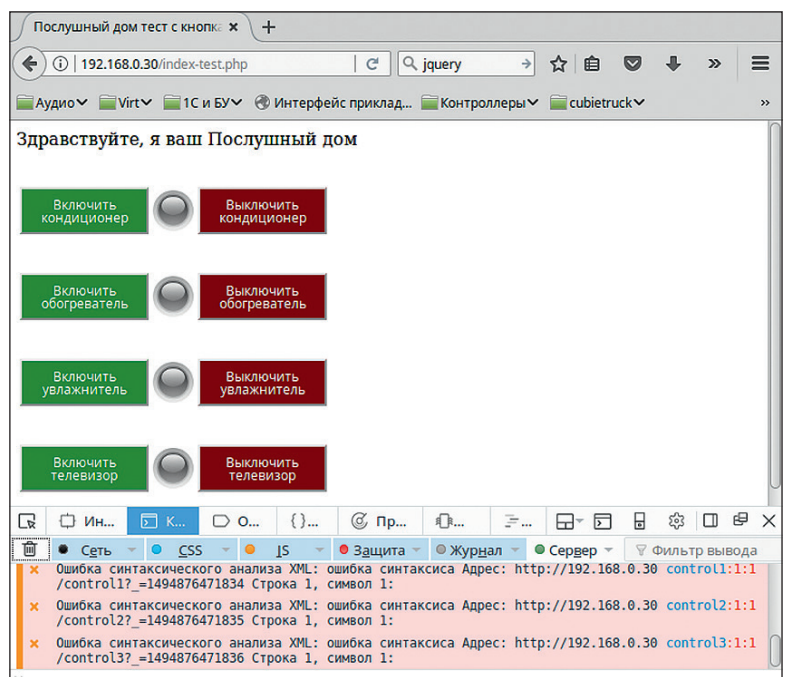
Причина проста: функция \$.ajax() узнает о типе присланных сервером данных от самого сервера (средствами MIME). Кроме этого, существует возможность лично указать (уточнить), как следует интерпретировать эти данные. Это делается с помощью параметра dataType. Возможные значения этого параметра:

- » “xml” — полученный XML-документ будет доступен в текстовом виде. С ним можно работать стандартными средствами jQuery (так же, как и с документом HTML).
- » “html” — полученный HTML будет доступен в текстовом виде. Если он содержит скрипты в тегах <script>, то они будут автоматически выполнены, только когда HTML-текст будет помещен в DOM.
- » “script” — полученные данные будут исполнены как JavaScript.

» “json”, “jsonp” — полученные данные будут предварительно преобразованы в JavaScript-объект. Если разбор окажется неудачным (что может случиться, если JSON содержит ошибки), то будет вызвано исключение ошибки разбора файла. Если сервер, к которому вы обращаетесь, находится на другом домене, то вместо json следует использовать jsonp.

» “text” — полученные данные окажутся доступными в виде обычного текста, без предварительной обработки.

Что удивительно, использование параметра dataType в допустимом варианте “text” в данной ситуации не помогает. На загрузку этот поток ошибок не влияет, да и обнаружил я его случайно. Тем не менее ошибки мы обязательно уберем, но чуть позже. LXF



► Рис. 2. Ошибки анализа XML.

MicroPython: Войны роботов

Лес Паундер показывает, как употребить две платы micro:bit и немного кода MicroPython на организацию битв радиоуправляемых роботов!



Наш эксперт

Лес Паундер — (джентльмен) хакер/разработчик, который любит возиться с Raspberry Pi, Arduino и micro:bit. Он готовит наставников для Raspberry Pi Foundation и описывает свои приложения на <http://bigl.es>.

Скорая помощь

Все коды проекта можно найти в нашем репозитории GitHub: <https://github.com/lesp/LXF222-micropython-robot/archive/master.zip>.

» Готовая тележка робота очень проста благодаря готовому шасси и драйверу мотора, созданному для micro:bit.

Мы познакомились с MicroPython в учебнике стр. 84 LXF221 и узнали, как, вдохновившись *Железным человеком*, создать себе перчатку, которая реагирует на данные в виде координат, взятых с акселерометра, встроенного в micro:bit. А на этом уроке мы построим кое-что получше!

Одним из самых популярных проектов для начинающих аппаратных хакеров является создание робота. Мы видели много разных вариантов, обычно употребляющих в качестве мозгов роботов различные модели Raspberry Pi, соединенные с платами управления мотором и шасси робота. Но с появлением micro:bit мы видим на рынке еще больше наборов для роботов и аксессуаров. Поскольку micro:bit не поставляется с Wi-Fi и не поддерживает Bluetooth с помощью MicroPython, как мы можем управлять роботом? Ответ заключается в том, что у нас есть простая радиосистема в BBC micro:bit, которую можно использовать для отправки коротких сообщений или данных.

На нашем уроке мы расскажем о следующих шагах по созданию и программированию робота:

- 1 Сборка шасси робота из комплекта.
- 2 Управление моторами с помощью специальной платы.

- 3 Создание блока контроллера, определяющего команды пользователя и интерпретирующего движение.

- 4 Установление радиосвязи между контроллером и роботом.

- 5 Настройка «мозга» робота для приема радиосигналов и соответствующего реагирования.

Подготовка к проекту

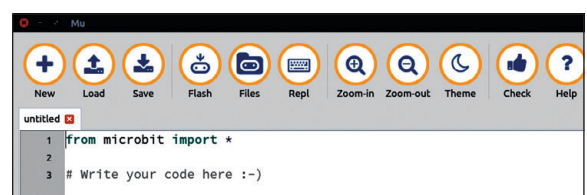
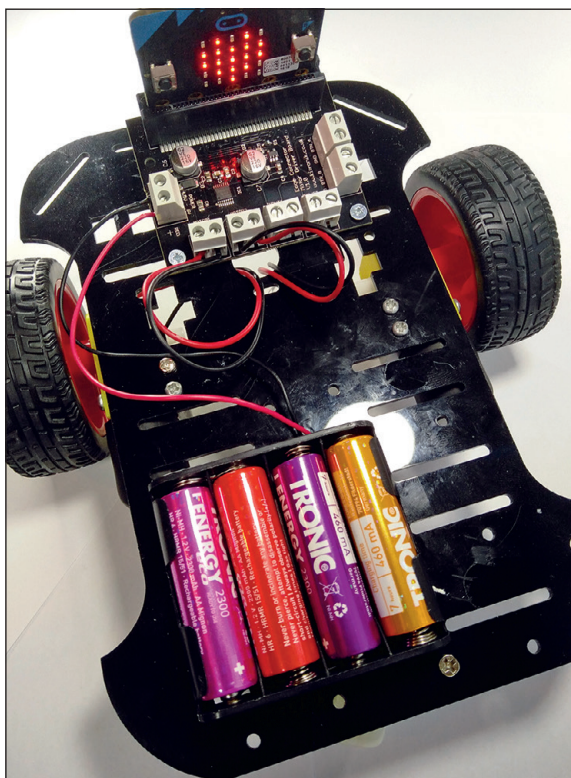
Для этого урока вам понадобятся два micro:bit (ведь для войны нужны две стороны), USB-аккумулятор, два кабеля USB — микро-USB, плата драйвера мотора Kitronik и комплект для роботов. Наборы для роботов по низкой цене можно найти на eBay; вам понадобятся такие, в которые входят два мотора, шасси, комплект из четырех батарей AA, два колеса и колесная тележка (с возможностью поворота переднего колеса).

Наш робот будет управляться рулем-баранкой от micro:bit. Этот руль будет считывать данные со встроенного акселерометра и передавать информацию другому micro:bit, установленному на роботе. Весь проект будет закодирован в MicroPython, в приложении *Mu*. MicroPython выпущен Python Software Foundation, он позволяет использовать более компактную реализацию Python 3; изначально MicroPython был создан Дэмиеном Джорджем [Damien George].

Используя MicroPython, два micro:bit, плату драйвера мотора и дешевое шасси робота, мы сделаем первые шаги в робототехнике с MicroPython и micro:bit. Установку *Mu* — редактора для MicroPython — мы рассмотрели в LXF221, а сейчас кратко напомним. Установка *Mu* на Linux машине проста: всё, что надо сделать — это загрузить приложение с https://s3-us-west-2.amazonaws.com/ardublockly-builds/microbit/linux/mu-2016-11-06_11_36_15, затем перейти в папку **Загрузки**. Щелкните по приложению правой кнопкой мыши и выберите Свойства, затем перейдите в Права и измените разрешения, чтобы файл стал исполняемым. Теперь можете дважды щелкнуть по приложению, и откроется редактор *Mu*.

Мы разбили урок на разделы, чтобы облегчить процесс изучения, поэтому давайте начнем собирать робота!

Наш проект использует два micro:bit: «мозг» робота и блок контроллера, используемый для отправки команд по радио. Начнем с кодирования контроллера. Для этого в *Mu* нажмем Создать,



» *Mu* — это редактор Micro Python для micro:bit. Он предлагает простой в использовании интерфейс, с автоматическими подсказками кода и отступами.

Радиосвязь в micro:bit

На нашем уроке для соединения двух micro:bit мы использовали радиосвязь, которая позволяла передавать строки данных между устройствами. Для простоты использовались настройки радио по умолчанию. А если мы хотим создать несколько роботов для битвы на арене? Именно здесь вступают в действие каналы. Класс radio имеет множество различных опций настройки, доступных пользователю, но канал — простейшая из них. Чтобы изменить канал радио, мы должны использовать:

```
import radio
radio.config(channel=99)
radio.on()

Теперь мы можем изменить канал от 0 до 100. Это надо проделать для каждого micro:bit, для которого вы хотите иметь определенный канал. Для большей дальности радиосвязи увеличьте мощность передачи. Это тоже настраиваемый параметр, он регулируется посредством следующей команды:
```

```
radio.config(power=7)

Мощность передачи варьируется от -30, -20, -16, -12, -8, -4, 0, 4 дБм, причем каждое из этих семи значений соответствует значению конфигурации от 0 до 7 соответственно. Если вы хотите узнать больше об отправке радиосигналов с помощью MicroPython, взгляните на его отличный официальный ресурс — сайт https://microbit-micropython.readthedocs.io/en/latest/radio.html.
```

чтобы начать с нового пустого документа. И не забывайте почаще сохранять:

```
from microbit import *
import radio
```

Кодирование контроллера

Сначала импортируем две библиотеки: всю библиотеку micro:bit, дающую доступ к датчикам и дисплею на плате, и radio, библиотеку, обеспечивающую ближнюю радиосвязь между micro:bit. Чтобы использовать радио, его сначала следует включить, это надо проделать для каждого micro:bit, используемого в проекте (см. также врезку «Радиосвязь в micro:bit» вверху для получения дополнительной информации):

```
radio.on()

Теперь прокрутим сообщение по светодиодной матрице контроллера micro:bit. Это сообщение пользователю о том, как управлять роботом. Мы также устанавливаем скорость прокрутки, используя задержку. Эта задержка будет определять частоту обновления матрицы. Значение по умолчанию — 150 мс, но мы уменьшили его до 50 мс, так как 150 мс было слишком медленно. display.scroll("Управляйте роботом, вращая руль",delay=50)
```

Для постоянной проверки входа драйвера воспользуемся бесконечным циклом; в MicroPython это

```
while True:

    Внутри цикла надо определить входные данные драйвера. Эти данные собираются проверкой показаний акселерометра. В LXF221 мы собирали и использовали отдельные координаты X, Y и Z, сгенерированные перемещением micro:bit, для управления светодиодами WS2812B. Но для этого проекта мы можем упростить данные акселерометра, используя «жесты». micro:bit умеет определять 11 жестов, таких как вверх, вниз, влево, вправо, тряска и т.д., а также может измерять ускорения, определяемые в G:
    gesture = accelerometer.current_gesture()
    print(gesture)
```

Здесь мы создаем переменную с именем gesture и сохраняем текущий жест. Мы также выводим содержимое нашей переменной gesture в оболочку Python REPL для отладки.

У нас есть хранящийся в переменной жест, и теперь мы проверим, соответствует ли он одному из тех, что нужны для управления роботом. Для этого используем тест if...elseif:

```
if gesture == "left":

    Этот первый тест проверяет, был ли контроллер повернут влево. Поэтому мы проверяем содержимое нашей переменной gesture на жестко закодированное значение "left".
    Итак, если этот жест значит «влево», необходимо дать водителю отклик для подтверждения:
    display.show(Image.ARROW_W)
```

```
radio.send('left')

Здесь мы обновили светодиодную матрицу на micro:bit контроллера, чтобы показать направление движения. Класс display содержит множество изображений, которые можно показать. Итак, для иллюстрации, что мы поворачиваем влево, мы обновляем дисплей для отображения стрелки, указывающей на «запад», что для micro:bit означает «влево». Радиосвязь поддерживает связь с «мозгом» робота, и по радио отправляется строка данных ожидающему «мозгу» робота. В строке указывается желаемое направление движения робота.
```

А если бы жест был «вправо»? Ну, здесь мы используем else if, известный в Python как elif. Elif будет проверяться, если первое условие вернуло False или предыдущие условия elif также вернули False. Мы продолжаем elif, пока не получим True. Поэтому, если мы сделали жест, что надо пойти вправо, запустится следующий код:


```
elif gesture == "right":
    display.show(Image.ARROW_E)
    radio.send('right')

То же повторяем при качании micro:bit вверх и вниз:
elif gesture == "up":
    display.show(Image.ARROW_N)
    radio.send('forward')
elif gesture == "down":
    display.show(Image.ARROW_S)
    radio.send('reverse')
```

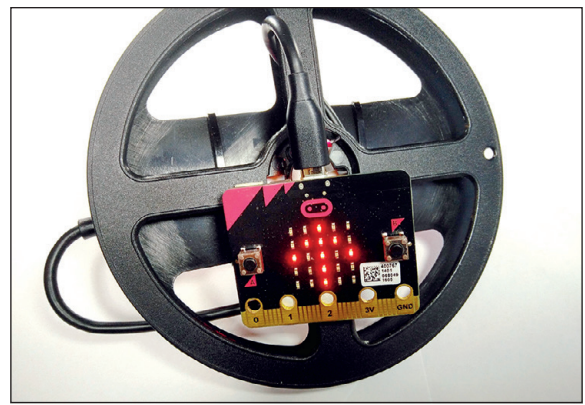
Нашим последним условием проверки elif является аварийный тормоз. На всякий случай, когда нужно, мы можем нажать кнопку B на micro:bit, чтобы остановить робота. Это работает, только если плата удерживается ровно!

```
elif button_b.was_pressed():
    display.show(Image.SURPRISED)
    radio.send('brakes')
```

Скорая помощь



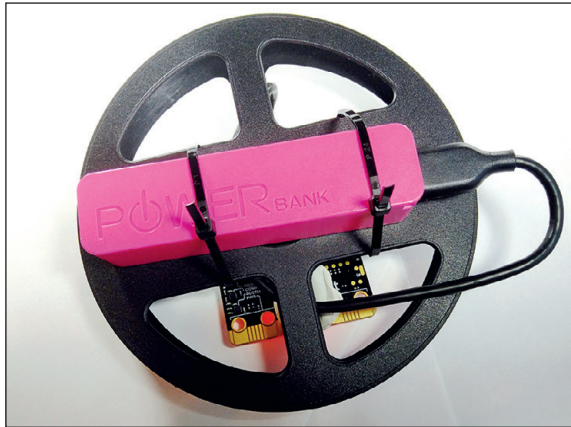
Правильное питание micro:bit может быть несколько мудренным. В идеале, он питается через разъемы JST и источника 3 В, но также доступен 5-В блок питания USB. Мы использовали дешевый блок питания из магазина «всё за фунт».



» Наш контроллер — это micro:bit, закрепленный на самодельном рулевом колесе с помощью пластилина Blu-Tack. Поворот micro:bit, как при управлении автомобилем, будет успешно управлять роботом.

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

Здесь мы использовали для питания `micro:bit` аккумуляторную батарею USB, но также сгодится и официальный батарейный комплект AAA на 3 В.



Вот и весь код для нашего контроллера. Позаботьтесь сохранить код на компьютере. Затем подключите `micro:bit` и нажмите Flash [Залить], чтобы загрузить код на `micro:bit`. Через несколько секунд вы увидите инструкции, прокручиваемые на светодиодной матрице. Теперь держите `micro:bit` как рулевое колесо, и вы увидите стрелку вверх, указывающую, что радио отправляет команду «вперед» нашему роботу. Поверните `micro:bit` влево и вправо, потом переверните вверх ногами для заднего хода. Наконец, держите `micro:bit` ровно и нажмите В, чтобы применить тормоз.

Чтобы получить более «реалистичное» ощущение управления роботом, мы использовали взамен рулевого колеса старое кольцо, используемое для прокладки кабелей и проводов по столбам. Несколько кабельных стяжек для надежного закрепления `micro:bit` и USB-аккумулятора, и мы готовы для поездки.

Сборка робота

Покончив с контроллером, перейдем к созданию нашего робота, и наша первая задача — собрать его корпус. Наш робот использует обычное шасси робота, найденное в онлайн-аукционе за примерно £ 10. Немного пайки, а именно — клеммы на каждом моторе. Это простая задача, но если вы не умеете паять или у вас нет набора, обратитесь за помощью в местный кружок Raspberry Jam или LUG.

Провода от моторов подключены к плате драйвера мотора Kitronik. Каждый мотор подключен к своим клеммам. Глядя на нашего робота спереди, мотор 1 находится слева от шасси, а мотор 2 — справа. Оба провода от одного мотора подключены к двум клеммам для мотора 1 на плате драйвера. Сделайте то же самое для второго мотора, убедившись что он подключен к мотору 2. Также убедитесь, что клеммы крепко удерживают провод. Теперь подключите блок батарей AA к клемме питания и проверьте на соответствие полярности. Обычно батарейный блок AA имеет красный и черный провода, где красный означает «+», а черный — «-». Батарейки пока не вставляйте. Теперь закрепите батарейный отсек, моторы и плату драйвера на шасси с помощью винтов из комплекта или — по-быстрому — кабельными стяжками. Теперь извлеките «мозг» робота `micro:bit` из шасси робота, поскольку мы должны его запрограммировать.

Используйте `micro:bit`, предназначенный для мозга нашего робота, а не `micro:bit` контроллера робота, который мы прошили раньше. Подключите `micro:bit` к компьютеру и запустите приложение *Mu*. Создайте новый файл, да не забывайте сохранять его почаще. Мы начинаем кодировать робота, импортируя те же библиотеки, что и для контроллера, а именно `microbit` и `radio`. Также обеспечим, что радио включено и готово к приему сигнала:

```
from microbit import *
import radio
```

```
radio.on()
```

Чтобы увидеть, что робот загружается и готовится к работе, мы создадим короткую анимацию, используя цикл `for`. В таких циклах применяется заданное количество итераций: например, в этом цикле `for` мы установим диапазон 3, что приведет к повтору цикла трижды:

```
for i in range(3):
```

По мере выполнения цикла мы отобразим на светодиодной матрице сердечко, и подождем полсекунды или 500 мс, используя функцию `sleep` Micro Python, для гарантии, что пользователь успеет рассмотреть изображение:

```
display.show(Image.HEART_SMALL)
```

```
sleep(500)
```

Затем заменим это изображение на большее изображение сердца и еще раз повторим сон, чтобы создать на светодиодной матрице эффект «сердцебиения».

```
display.show(Image.HEART)
```

```
sleep(500)
```

Кодирование робота

Написав код запуска робота, мы переходим к основному циклу, который отвечает за постоянную проверку радиосигналов. Мы опять используем бесконечный цикл, содержащий необходимый код:

```
while True:
```

Внутри цикла мы создаем новую переменную, дав ей имя `incoming`; она будет хранить входящие радиосообщения, отправленные нашим контроллером `micro:bit`.

```
incoming = radio.receive()
```

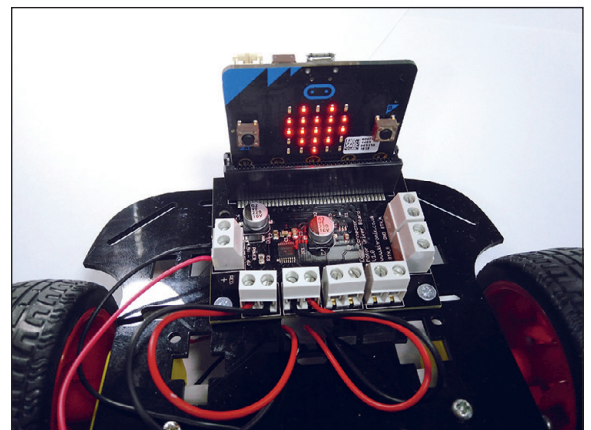
Чтобы интерпретировать и обрабатывать команды, отправляемые по радио, наш робот должен сначала понять, что делать по конкретной команде, и здесь мы снова используем условный тест `if...elif`. Мы проверяем, что значение нашей входящей переменной совпадает с жестко закодированным значением, например, 'left':

```
if incoming == 'left':
```

Если была команда повернуть налево, мы обновим светодиодную матрицу `micro:bit` мозга нашего робота, чтобы показать направление движения. В этом случае мы обновляем стрелку, указывающую на восток, так как при взгляде на робота в лоб стрелка, указывающая на восток, также направлена влево.

```
display.show(Image.ARROW_E)
```

Для управления моторами необходимо управлять выводами GPIO, к которым они подключены через плату драйвера мотора. Как можно видеть, на плате драйвера мотора рядом с контактами есть номера 'P'. Это выводы GPIO, подключаемые к контактам.



У нашего робота есть мозги и сердце! Анимация сердца показывает, что робот готов к работе.

Скорая помощь

Различных типов моторов много. Мы использовали стандартные моторы постоянного тока, но вы также можете найти дешевые микродвигатели с редуктором, позволяющие построить робота меньшего размера.

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

Для работы мотора в одном направлении на одном выводе должен быть высокий уровень, а другом — низкий. Другими словами, может быть включен только один вывод, а другой должен быть выключен. Чтобы изменить направление вращения мотора, необходимо изменить эту полярность:

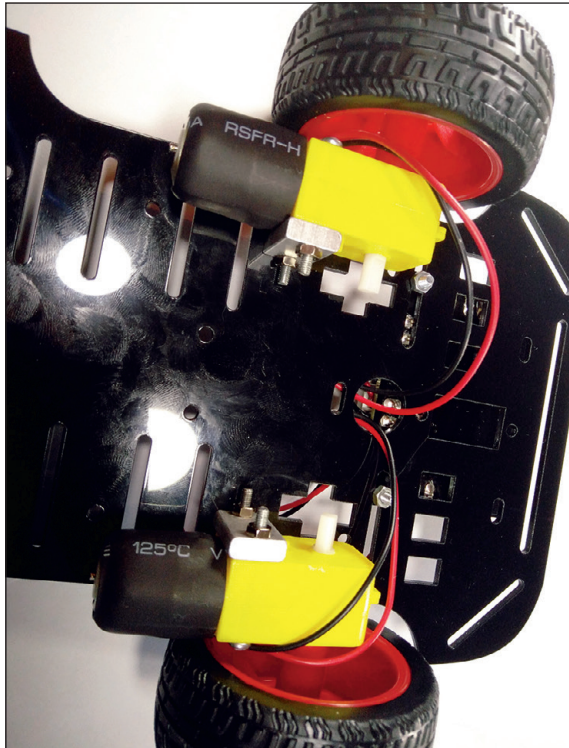
```
pin8.write_digital(0)
pin12.write_digital(1)
pin16.write_digital(0)
pin0.write_digital(1)
```

Управление мотором

Чтобы повернуть робота влево, надо велеть одному мотору вращаться вперед, а другому — назад. В нашем примере первый двигатель — левый (если смотреть на робота спереди), и он подключен к выводам 12 и 8. Наш правый мотор — это второй двигатель, управляемый выводами 16 и 0. Поэтому, чтобы повернуть робота влево, первый мотор должен вращаться вперед, а второй — назад. Мы делаем это, подавая питание на соответствующие выводы GPIO, используя `write_digital`.

Код поворота направо противоположен коду поворота налево. Вот демонстрация, что схема может легко переключать полярность «на лету»:

```
elif incoming == 'right':
    display.show(Image.ARROW_W)
    pin8.write_digital(1)
    pin12.write_digital(0)
    pin16.write_digital(1)
    pin0.write_digital(0)
    Вот код для движения робота вперед:
elif incoming == 'forward':
    display.show(Image.ARROW_N)
    pin12.write_digital(1)
    pin8.write_digital(0)
```



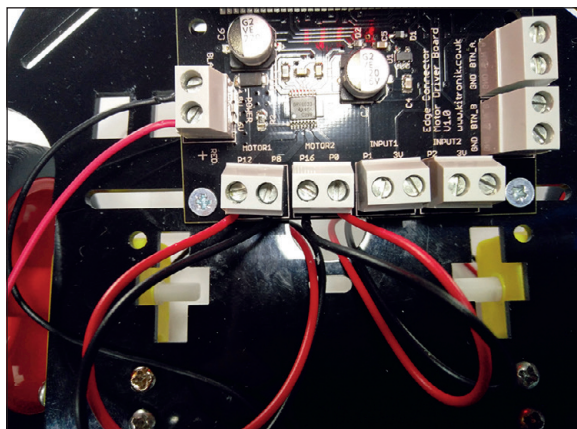
➤ На нашем шасси имеются всевозможные отверстия для крепления компонентов, но моторные пилоты могут быть только в этом месте, для точного баланса, необходимого нашему роботу.

```
pin16.write_digital(1)
pin0.write_digital(0)
    А вот код для движения робота назад:
elif incoming == 'reverse':
    display.show(Image.ARROW_S)
    pin12.write_digital(0)
    pin8.write_digital(1)
    pin16.write_digital(0)
    pin0.write_digital(1)
```

Наше последнее условие теста — аварийный тормоз. Оно останавливает все выводы GPIO, подключенные к моторам, в ноль:

```
elif incoming == 'brakes':
    display.show(Image.SURPRISED)
    pin12.write_digital(0)
    pin8.write_digital(0)
    pin16.write_digital(0)
    pin0.write_digital(0)
```

Сохраните код и нажмите Flash, чтобы загрузить код на `micro:bit`. Теперь отключите `micro:bit` от компьютера и установите его в слот платы драйвера мотора. Вставив батарейки, вы должны увидеть начало анимации сердцебиения. Теперь подайте питание на контроллер; мы использовали USB-батарею телефона. После прокрутки инструкции вы сможете управлять своим роботом в комнате. Если робот ведет себя немного иначе, вам может понадобиться поменять местами провода на клеммах от каждого мотора. **LXF**



➤ Двигатель 1 — мотор слева на рисунке, а двигатель 2 — справа. Можно также видеть провода в пространстве шасси, идущие к моторам.

Зачем нужен контроллер

На нашем уроке мы использовали плату драйвера мотора для работы `micro:bit` с двумя моторами, но зачем нам контроллер? А просто для буферизации `micro:bit`, поскольку моторы требуют значительного тока для начала движения.

Этот ток `micro:bit` не может обеспечить без потенциального повреждения устройства. Мы использовали плату драйвера мотора Kitronik, но в ее основе

лежит контроллер мотора DRV8833, использующий схему H-мост или триггер для управления направлением тока. Помните, что в уроке мы изменили полярность моторов, чтобы изменить направление? Так вот это работа схемы H-моста.

А надо ли покупать комплект Kitronik? Вы можете сделать свой собственный, купив плату управления мотором, такую как DRV8833, L293D или L9110S;

все они совместимы с `micro:bit`, но вам придется выделить больше выводов GPIO `micro:bit`, трех имеющихся недостаточно.

Вы можете приобрести переходники к разъему `micro:bit` для использования с макетными платами и обычной электроникой, это снизит стоимость ваших роботов и позволит создать кучу роботов для вашей новой армии!



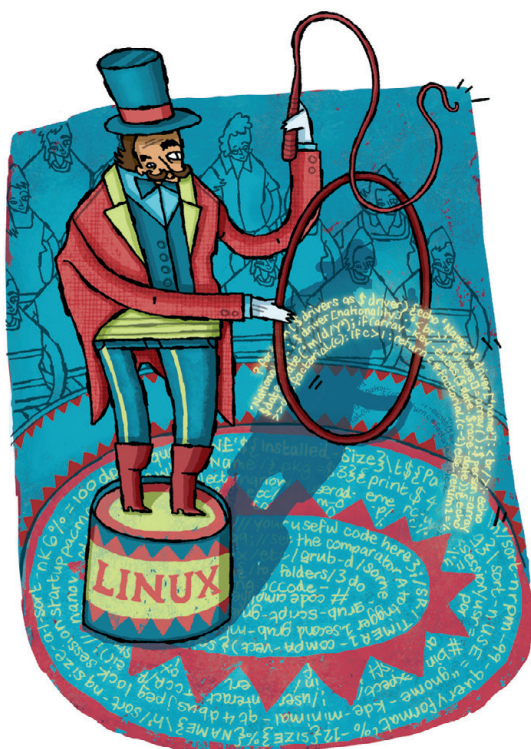
Django: Изучим фреймворк

Томас Рамболд рассказывает об основах фреймворка Django, а Дэниел Сэмюэлс показывает, как написать первые строки кода.



Наш эксперт

Томас Рамболд занимался web-разработкой, разработкой прикладных программ и был техническим писателем. А сейчас он технический директор агентства Onepersamedia, которое специализируется в web-дизайне.



Web-фреймворк (или фреймворк web-приложений) — это программная среда, предназначенная для поддержки разработки сайтов и web-приложений, включая web-службы, web-ресурсы и web-интерфейсы API. Благодаря web-фреймворкам снижаются накладные расходы, которые связаны с рутинными действиями, выполняемыми при web-разработке. С этой целью предоставляется набор инструментов, предназначенных для упрощения и ускорения процесса разработки для web-разработчиков.

Итак, что же такое Django и в чем он хорош? По-простому, Django — это web-фреймворк Python, позволяющий разработчикам сайтов и web-приложений быстро и надежно создавать сложные проекты, а также поддерживать согласованный «чистый» стиль. Django содержит ряд готовых инструментов, которые поддерживают, улучшают и ускоряют традиционный процесс web-разработки и позволяют реализовывать сложные проекты быстро и по очень высоким стандартам.

Django предоставляет набор готовых программных инструментов, которые берут на себя большую часть задач, не связанных напрямую с проектом. А значит, разработчики смогут потратить больше времени на написание важных вещей (например, самих функций), в то время как обычные, стандартные действия

по обслуживанию web-приложения будут выполняться автоматически в фоновом режиме.

Возможно, вы спрашиваете себя: «А слышал ли я о каких-то проектах, написанных на Django?». Да, пожалуй. Pinterest, платформа для создания подборок изображений, на самом деле представляет собой распределенную систему на Django с очень большим трафиком. Web-приложение Instagram тоже написано на Django. На сайтах *The Washington Post*, *The Onion* и *NASA* также используется Django. Возможно, вы уже почуяли идею. Эти ребята отвечают за сложные, крупномасштабные системы с очень большим трафиком — и делают это неплохо (по крайней мере, на первый взгляд). Bitbucket и EventBrite тоже основаны на Django (и на самом деле, в состав команды EventBrite входят несколько человек из основной группы разработчиков Django).

И как же это выглядит с технической стороны? В Django есть масса великолепных возможностей и отличных функций. Описать все возможности Django в одной статье было бы невозможно (и для этого существует документация), но в целом с технической точки зрения в Django приходится иметь дело с пятью основными компонентами.

Приложения

Объединив различные функции Django, вы создаете приложение, которое выполняет одну функцию и делает это очень хорошо. Приложение в Django — это автономная структура кода, состоящая из нескольких файлов, которые обычно содержат все необходимое для работы приложения, а также ссылки на другие приложения, если вам надо обмениваться данными между ними. Приложение состоит из ряда базовых файлов. Вы также можете создавать новые файлы и при необходимости импортировать, но для простых приложений достаточно базовой структуры проекта.

Модели

Модели в Django — это механизм, позволяющий разработчикам взаимодействовать с базой данных (создавать, считывать, обновлять и удалять данные), не взаимодействуя с уровнем базы данных напрямую. Модели в мире программирования рассматриваются как уровень абстракции между базой данных и уровнем приложения — а в конкретной реализации Django это означает, что вам почти никогда не придется писать SQL-запросы, так как ваша модель является полнофункциональным Python-представлением таблицы базы данных. Это означает, что вместо написания SQL-запроса вы можете просто импортировать и вызывать элементы вашей модели в Python, не опускаясь ниже уровня приложения.

Это блестящий способ сохранить чистоту и хорошую структурированность кода и использовать его повторно, а также ограничить любые потенциально опасные или неуместные взаимодействия. Это также означает, что многие данные и операции,

Скорая помощь



С помощью виртуальных сред можно изолировать зависимости в проектах. Чтобы запустить виртуальные среды, выполните команду `pip install virtualenv virtualenv`.
`.venv . .venv/bin/activate`.

Для чего служит каждый из этих файлов?

При запуске `startproject` создается четыре файла: `manage.py`, `settings.py`, `urls.py` и `wsgi.py`.

Файл `manage.py` — это замена команды `django-admin`, которой мы пользовались ранее. Это основная точка входа для вашего приложения, и этот файл используется для запуска любых команд управления, связанных с нашим проектом (например, создание

пользователя-администратора). Изменять этот файл приходится чрезвычайно редко.

Файл `settings.py` содержит все параметры нашего проекта, такие как имя проекта, URL-адрес проекта и т.д. Мы не будем изменять этот файл, но в него стоит заглянуть и посмотреть, что там содержится.

Файл `urls.py` — базовый файл маршрутизации для всех запросов, которые поступают в Django.

Он сопоставляет различные URL-адреса с соответствующими представлениями. Мы изменим этот файл позже.

Файл `wsgi.py` используется при развертывании проекта на сервере. Он позволяет HTTP-серверу WSGI — например, такому как Gunicorn — подключить ваше приложение и передавать ему HTTP-запросы.

связанные с SQL (такие как проверка), можно хранить и выполнять на уровне модели. Следовательно, просканировав файл модели, можно понять, как этот файл соответствует таблицам в вашей базе данных (и всё это благодаря Django).

Представления

Создав модель, создав базу данных и сохранив в ней некоторые данные, вы, вероятно, захотите что-то сделать с этими данными. Для передачи данных в HTML-шаблон для последующего просмотра пользователем необходимо так называемое «представление» — это файл, предназначенный для обработки и передачи данных в другие компоненты web-приложения. Представления очень эффективны, так как это всего лишь файлы Python. С представлениями можно делать всё то, что можно делать с файлами Python — обращаться, разделять и объединять данные практически любым способом. В Django есть целый набор инструментов, которые помогут разработчикам быстро это сделать. Представления Django на основе общих классов позволяют экономить время, так как вам не приходится снова и снова писать один и тот же код для разных проектов.

URL-адреса

В основе любой хорошей web-архитектуры лежат хорошие системы формирования URL-адресов, и Django позволяет легко создавать простые и динамичные системы формирования URL-адресов. С каждым приложением обычно связан файл `urls.py`, позволяющий задать набор маршрутов, с которым Django будет сравнивать текущий URL-адрес посетителя сайта. Если совпадение обнаружено, Django вызовет связанное с этим URL-адресом представление и выполнит все связанные с ним действия, возвратив результат в соответствующий шаблон, а пользователя — на соответствующий URL-адрес. Эта встроенная система обработки URL-адресов очень эффективна для создания больших динамических схем формирования URL-адресов с минимальными издержками.

Шаблоны

Шаблоны — это компоненты сайта или web-приложения, которые выступают в качестве отдельных HTML-страниц или в качестве компонентов других HTML-страниц для посетителей сайта. Шаблоны создаются на языке HTML, и их стиль определяется CSS — но благодаря собственному языку шаблонов Django можно легко вызывать объекты базы данных. В шаблоны можно передать данные из соответствующего файла представления. Данные шаблонов можно передать в представление через форму ввода.

Синтаксис шаблонов Django прост и эффективен, но рекомендуется как можно лучше изолировать логику от приложений, чтобы код оставался чистым и понятным. Любую сложную обработку или фильтрацию следует выполнять на уровне модели или представления (или, возможно, в теге шаблона). Поэтому на момент загрузки

в фактическом шаблоне обработка уже завершена, и работа с клиентской частью максимально упрощается. В Django много интересных встроенных инструментов фильтрации, которые помогут обработать ваши данные — на них стоит обратить внимание.

Администрирование

Автоматическая система администрирования Django очень полезна — в основном благодаря времени, которое она экономит. Она работает так: у любого создаваемого вами приложения есть собственный файл администрирования. Для создания системы администрирования достаточно импортировать в этот файл соответствующие модели базы данных и затем выбрать визуальное расположение этих полей. Django возьмет эти компоненты и сделает их доступными для администратора (по умолчанию — с использованием URL-адреса `/admin`). Удобство этого подхода состоит в том, что если вам не нужны сложные функции, то и делать ничего не надо. Вы создаете очень маленькие файлы администрирования, и система администрирования готова. Подумайте, как долго вы создавали бы эту систему самостоятельно?

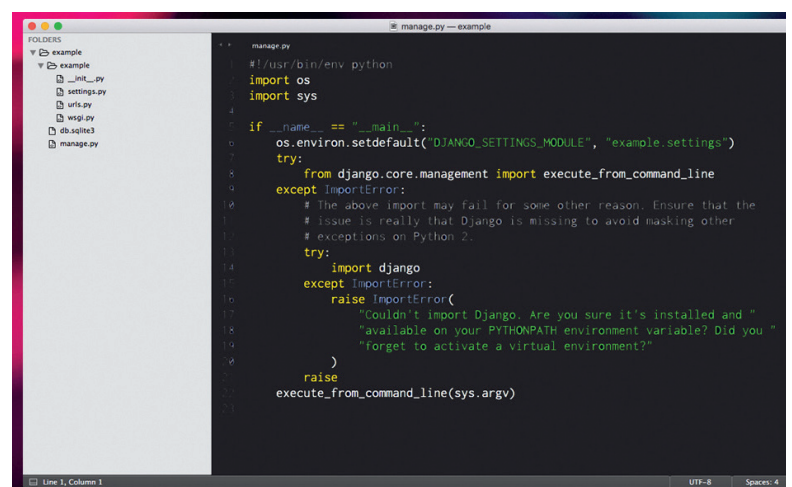
Начинаем

Сообщество Django очень активно и включает высококвалифицированных разработчиков. И если вы захотите начать работу с Django, то получите в свое распоряжение множество ресурсов. Чтобы продемонстрировать простоту Django, мы создадим простое приложение для ведения блога. (Окончательная версия исходного кода доступна здесь: <https://github.com/danielsamuels/gcbv-article>.)

Для начала вам надо установить Python. В OS X и Linux он уже установлен, так что вы в дамках. В Windows необходимо перейти

Скорая помощь

В Python есть полезная программа для установки зависимости, под названием `pip`. Для установки Django выполните команду `pip install Django`. Если вы хотите экспортировать свои требования, выполните команду `pip freeze > requirements.txt`, а другой пользователь сможет установить эти требования, скопировав `pip install -r requirements.txt`.



```

manage.py
#!/usr/bin/env python
import os
import sys

if __name__ == "__main__":
    os.environ.setdefault("DJANGO_SETTINGS_MODULE", "example.settings")
    try:
        from django.core.management import execute_from_command_line
    except ImportError:
        # The above import may fail for some other reason. Ensure that the
        # issue is really that Django is missing to avoid masking other
        # exceptions on Python 2.
        try:
            import django
        except ImportError:
            raise ImportError(
                "Couldn't import Django. Are you sure it's installed and
                'available' on your PYTHONPATH environment variable? Did you
                forget to activate a virtual environment?"
            )
        raise
    execute_from_command_line(sys.argv)

```

» Структура проекта Django по умолчанию. Это шаблон, который Django предоставляет вам для использования. И это отличная отправная точка.

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



на сайт Python, чтобы загрузить и установить Python. Перед установкой Django мы создадим так называемую «виртуальную среду». Виртуальная среда позволяет изолировать зависимости проекта от остальной системы, чтобы вы не оказались в ситуации, когда одновременно загружается несколько версий одной и той же библиотеки. Чтобы начать использовать виртуальную среду в Python 2, надо установить пакет: `pip install virtualenv`. В Python 3 виртуальная среда уже есть, и мы просто воспользуемся ею.

Чтобы создать виртуальную среду в Python 2, скомандуйте `virtualenv .venv`, а в Python 3 введите команду `python3 -m venv .venv`. После создания среды ее надо активировать перед использованием. Для этого скомандуйте `./venv/bin/activate`.

Теперь мы готовы к установке Django (так, чтобы она не повлияла ни на что другое). Для этого выполните команду `pip install Django`. Загрузится и установится последняя версия Django. Чтобы приступить к использованию Django, для начала надо создать проект. Для создания проекта воспользуемся базовым шаблоном Django. Выполните команду `django-admin startproject blog`. Заглянув в этот каталог, вы увидите, что было создано несколько файлов, которые станут основой нашего проекта.

Получив первые файлы проекта, приступим к работе над ним. Чтобы сохранить рабочее пространство чистым, мы будем создавать файлы приложений в каталоге проекта, а не вместе со всеми остальными файлами, поэтому создайте каталог под названием **blog**. Также понадобится создать в этом каталоге пустой файл с именем `__init__.py`, чтобы Python знал, что это загружаемый модуль. После этого надо создать в этом каталоге файл `models.py`, который будет содержать наше представление записи в блоге. Оставим его простым, пока включив только название и содержимое.

```
from django.db import models
```

```
class Post(models.Model):
```

```
    title = models.CharField(
        max_length=100,
    )
```

```
    content = models.TextField()
```

Затем нам понадобятся отдельные представления для получения запроса, определения необходимых действий и возврата веб-страницы. Для этого мы будем использовать представления Django на основе общих классов, которые позволяют легко решать пространственные задачи. Создайте файл `views.py` со следующим содержанием в каталоге вашего блога:

```
from django.core.urlresolvers import reverse
```

```
from django.views.generic import CreateView, DeleteView,
```

```
DetailView, ListView, UpdateView
```

```
from .models import Post
```

```
class PostListing(ListView):
```

```
    model = Post
```

```
class PostCreate(CreateView):
```

```
    model = Post
```

```
    success_url = '/'
```

```
    fields = ['title', 'content']
```

```
class PostDetail(DetailView):
```

```
    model = Post
```

```
class PostUpdate(UpdateView):
```

```
    model = Post
```

```
    fields = ['title', 'content']
```

```
    def get_success_url(self):
```

```
        return reverse('blog:detail', kwargs={
```

```
            'pk': self.object.pk,
```

```
        })
```

```
class PostDelete(DeleteView):
```

```
    model = Post
```

```
    success_url = '/'
```

Последний файл в каталоге приложения, который нам потребуется — `urls.py`. Этот файл сопоставит URL-адрес в браузере с соответствующим представлением в файле `views.py`. Для проверки некоторых параметров URL-адреса мы будем использовать регулярные выражения, но всё, что они делают — находят число и передают его в представление в виде значения параметра «pk», то есть первичного ключа, который будет использоваться для поиска записи блога в базе данных.

```
from django.conf.urls import url
```

```
from .views import PostCreate, PostDelete, PostDetail,
```

```
PostListing, PostUpdate
```

```
urlpatterns = [
```

```
    url(r'^$', PostListing.as_view(), name='listing'),
```

```
    url(r'^create/$', PostCreate.as_view(), name='create'),
```

```
    url(r'^?P<pk>+d+/$', PostDetail.as_view(), name='detail'),
```

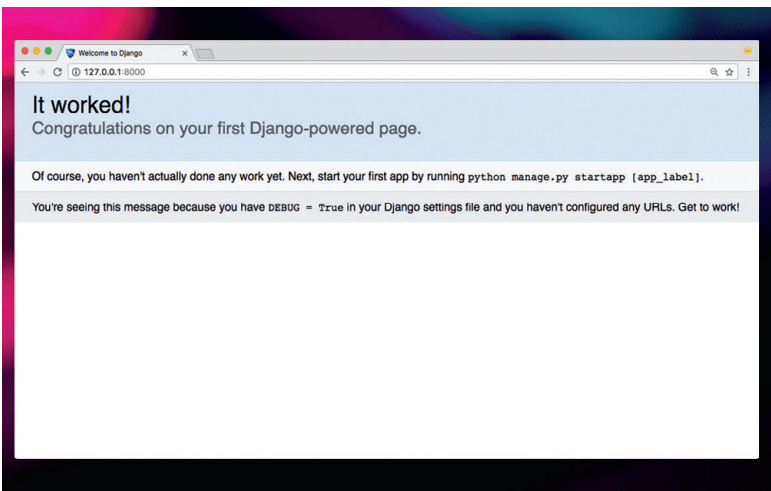
```
    url(r'^?P<pk>+d+/update/$', PostUpdate.as_view(),
```

```
        name='update'),
```



Скорая помощь

Проекты Django работают лучше всего, когда они разделены на небольшие части, поэтому подумайте о том, чтобы разделить свой проект на отдельные приложения, а затем разделите отдельные компоненты приложения на отдельные файлы. Поддерживать такие проекты гораздо проще.



➤ Это страница, которая откроется после первого запуска сервера разработки и перехода по URL-адресу <http://127.0.0.1:8000>.

Абстрагируемся от сложности

Приведенный выше пример кода показывает, как легко разрабатывать приложения на Django. Меньше 30 строк кода — и вы можете выводить список новых статей, создавать новые статьи, просматривать, обновлять и удалять их.

Это стало возможным благодаря тому, что разработчики Django потратили много времени

на создание коллекции общих представлений, которые являются удобными, но также и абстрактными. Они предоставляют вам всю необходимую функциональность и не мешают вам — а если вам не нравится, как в них что-то реализовано, вы можете просто переопределить или расширить их с помощью своих методов.

Этот уровень мышления распространяется на весь спектр функций Django, с акцентом на простоту и безопасность. Зная, что фреймворк аккуратно проделает за вас всю трудную работу, вы можете заниматься исключительно логикой приложения, работая более продуктивно и решая задачи своевременно.

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

```
url(r'^(?P<pk>\d+)/delete/$', PostDelete.as_view(), name='delete'),
]
```

Прежде чем интегрировать приложение в наш проект, надо сделать еще кое-что. Сначала добавим приложение в список установленных приложений в параметрах. Откройте файл `settings.py` и найдите `INSTALLED_APPS`. Добавьте еще одну строку со значением `blog.blog` ("blog" — название каталога проекта, а также название каталога приложения). Последнее, что надо сделать — добавить URL-адреса блога в URL-адреса проекта. Откройте файл `urls.py` в корневом каталоге проекта и добавьте следующую строку под строкой `admin`:

```
url(r'^', include('blog.blog.urls', namespace='blog'))
```

Также надо добавить `include` в список импортов из `django.conf.urls`:

```
from django.conf.urls import include, url
```

Потом следует сообщить Django о новой модели статей, чтобы Django узнал: в ней надо создать таблицу базы данных. Перейдите в терминал и выполните команду `python manage.py makemigrations blog`. Вы увидите, что команда создала файл. Затем скомандуйте `python manage.py migrate`, и все модели проекта превратятся в таблицы базы данных. Теперь всё готово для запуска сервера разработки. Выполните команду `python manage.py runserver`. Вы увидите, что сервер запустился, и на него можно перейти по ссылке <http://127.0.0.1:8000>. Вы получите сообщение об отсутствии шаблона, и это хорошо: это означает, что вы открыли правильное представление (в данном случае, представление `PostListing`).

Затем надо добавить все необходимые шаблоны. Их потребуется всего четыре: список статей блога, сведения о статье, формы и подтверждение удаления. Мы намерены сделать их как можно проще, поэтому давайте создадим следующие файлы:

» `templates/blog/post_confirm_delete.html`

```
<form action="" method="post">
  {% csrf_token %}
  <input type="submit">
</form>
```

» `templates/blog/post_detail.html`

```
<h1>{{ post.title }}</h1>
<p>{{ post.content }}</p>
```

» `templates/blog/post_form.html`

Onespacemedia.com

Onespacemedia — это агентство из Кембриджа по проектированию и разработке сайтов, специализирующееся на Python и работающее с наиболее инновационными компаниями в области науки и техники. Мы используем Django для большинства сайтов и приложений.

Наши недавние проекты включают финансовое приложение для MoneyMover, которое дало пользователям возможность

проводить торги с общим объемом более €100 млн; платформу для онлайн-обучения на дому COSMOS в сфере детского научного образования, которым занимается благотворительное учреждение Cambridge Science Centre; а также текущие проекты для Imperial Innovations, которые гарантируют, что их цифровое присутствие отражает их статус победителей в области инноваций.


```
<form action="" method="post">
  {% csrf_token %}
  {{ form.as_p }}
  <input type="submit">
</form>
```

» `templates/blog/post_list.html`

```
<ul>
  {% for post in post_list %}
  <li><a href='{% url "blog:detail" post.pk %}'>{{ post.title }}</a></li>
  {% endfor %}
</ul>
```

Окончательная структура файлов будет выглядеть так:

```
__init__.py
models.py
urls.py
views.py
templates/blog/post_confirm_delete.html
templates/blog/post_detail.html
templates/blog/post_form.html
templates/blog/post_list.html
```

Обновив окно браузера, вы увидите, что сайт работает (может потребоваться перезапустить сервер разработки). Ваш первый блог на Django готов! 

```
(example) DS:example danielsamuels$ ./manage.py runserver
Performing system checks...
```

```
System check identified no issues (0 silenced).
```

```
April 05, 2017 - 09:27:20
```

```
Django version 1.10.6, using settings 'blog.settings'
```

```
Starting development server at http://127.0.0.1:8000/
```

```
Quit the server with CONTROL-C.
```

» В Django есть встроенный сервер, который можно использовать при разработке приложений. С ним очень легко развернуть и запустить собственный проект.



Web-разработка: Инструментарий

Кент Ельчук объясняет, как войти в web-разработку задешево.



Наш эксперт

Кент Ельчук поработал профессиональным писателем и web-разработчиком, и увлекался различными хобби. Он создавал и простые сайты, и сложные системы с несколькими web-приложениями для колледжей.

Одна из основных причин использовать Linux в том, что он упрощает web-разработку. Говоря это, мы подразумеваем, что в Linux легко и бесплатно создавать и тестировать сайты и web-приложения.

Поскольку на серверах Linux размещено множество сайтов, включая таких гигантов, как Google и Facebook, применение Linux для web-разработки позволяет получить надежную среду для разработки и тестирования.

Если это вас не убедило, возможно, вас убедит статья <https://en.wikipedia.org/wiki/Goobuntu>: там сказано, что 10000 сотрудников Google пользуются Goobuntu. Это разновидность Ubuntu; следовательно, Ubuntu — хорошее решение.

Мы будем создавать среду для web-разработки с нуля. Это можно проделать на настоящем компьютере или в *VirtualBox*, выбор за вами. Независимо от этого выбора, помните, что технологии web-серверов со временем меняются, поэтому на *VirtualBox* вы сможете запускать последние серверные продукты и технологии, такие как *Apache*, *PHP*, *mysql* и *PhpMyAdmin*.

Начинаем с нуля

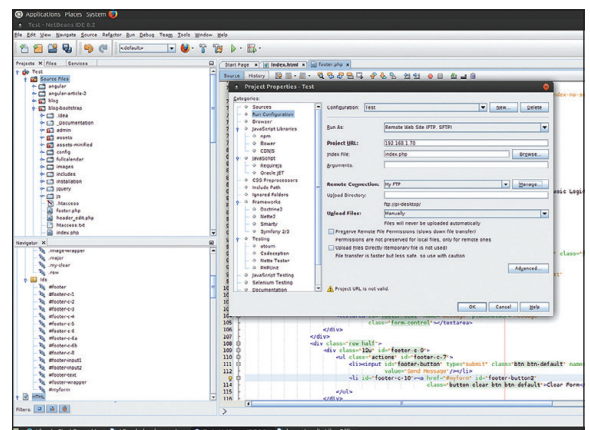
Выбирая для web-разработки настоящий компьютер, можете взять новый или бывший в употреблении. У новых есть очевидные плюсы, но среди б/у тоже отыщется немало хороших вариантов. Сейчас на интернет-аукционе можно купить старый ноутбук Lenovo бизнес-класса и рабочие станции HP XW8600 дешевле чем за \$100. Подключите к такому твердотельный жесткий диск — и вот вам умелый ПК для разработки.

Следует также учитывать свой дистрибутив Linux; если вы намерены использовать Raspberry Pi или хотите обеспечить совместимость с Raspberry Pi, основной вариант — Ubuntu Mate. Что касается самого Pi — рекомендуется использовать Raspberry Pi 3: эта модель работает намного быстрее по сравнению с предыдущими, и вы сможете подключить телевизор через вход HDMI, мышь и клавиатуру для настольного использования или просто запустить web-сервер на Raspberry Pi и перенести файлы на него. Таким образом, его можно использовать как удаленный web-сервер по аналогии с VPS или выделенным сервером от провайдера.

Программное обеспечение

Для web-разработки на выбранном компьютере вам понадобится установить web-сервер, базу данных и дополнительные скриптовые языки. Следуя данным Web, на нашем уроке мы настроим традиционный стек LAMP: Linux, *Apache*, *mysql* и PHP (или Perl либо Python). На нашем уроке P будет означать PHP. Если верить статистике на странице <https://w3techs.com/technologies/details/pl-php/all/all>, PHP используется на 82,5% сайтов, у которых язык программирования указан, так что мы в хорошей компании.

Статистика не означает, что PHP обязательно лучше остальных языков, и немного поискав информацию об этом в Google, вы узнаете, что, похоже, у PHP есть целая армия противников. Однако



► Настройте FTP с Netbeans для передачи файлов, и вы сможете легко получить компьютер для разработки и web-сервер в одном флаконе.

благодаря его популярности появилось огромное количество инструментов, благодаря которым с PHP легко начать работать (например, Wordpress — самая популярная CMS в мире).

Перейдем к настройке LAMP. Последовательность команд приведена ниже. При появлении запросов с вариантами 'yes' и 'no' без долгих раздумий выбирайте 'yes'.

Пароли для *mysql* и *PhpMyAdmin* можно оставить пустыми, завершая установку. Это нормально при настройке тестовой среды на платформе Linux, но если вы планируете использовать сервер в качестве web-узла с перенаправлением портов на роутере, следует установить надежные пароли и сохранить их в безопасном месте (если вы не уверены, что вдруг не забудете их).

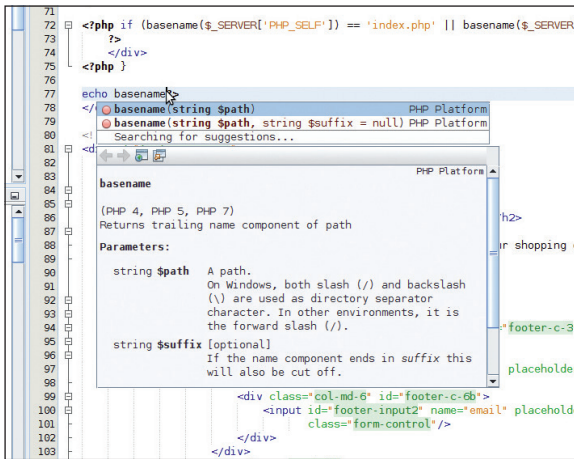
```
sudo apt-get update
sudo apt-get install apache2
sudo apt-get install mysql-server
sudo apt-get install php7.0 php-pear
libapache2-mod-php7.0php7.0-mysql
sudo apt-get install php7.0-curl php7.0-json php7.0-cgi
sudo apt-get install phpmyadmin
<Press Enter a few times during installation [Во время установки несколько раз нажмите Enter]>
```

В современных системах *PhpMyAdmin* не будет работать корректно, если вы захотите использовать его в тестовой среде без пароля. Подробности из первых уст можно получить на странице <https://docs.phpmyadmin.net/en/latest>. Если вы хотите заниматься web-разработкой сайтов с использованием *mysql* (к таким сайтам относятся сайты на Wordpress) и обращаться к базе данных с помощью графического интерфейса, без *PhpMyAdmin* не обойтись.

На данном этапе нам надо добавить еще один компонент для установки. Это *Samba*. Этот сервер можно установить и на гостевой компьютер, и на хост. Если у вас есть еще один компьютер с Linux,

Скорая помощь

Срок жизни Windows Vista закончился в апреле 2017 г., так что вы сможете приобрести компьютер с этой системой очень дешево. Один из источников хороших компьютеров бизнес-класса — правительственные аукционы. Выберите компьютер с 4 Гб оперативной памяти и подключите к нему твердотельный жесткий диск.



Версия Netbeans для разработчиков PHP с дополнением кода и подсказками с описанием функций и их параметров.

например, Raspberry Pi, где вы хотите сохранять файлы или делать резервные копии, *Samba* позволит это сделать — вы сможете перетаскивать файлы и каталоги, как на обычном ноутбуке или настольном компьютере. Если вы не хотите устанавливать *Samba* и создавать резервные копии на сервере *Samba*, всё связанное с *Samba* можете просто пропустить.

Для установки *Samba* и ее базовой настройки выполните такие команды:

```
apt-get install samba samba-common-bin
service samba start
или
/etc/init.d/samba start
```

Выполнив следующие команды, вы создадите каталог для обмена файлами, */home/mysamba*:

```
cd
mkdir kent
chown kent:sambashare kent
cd /etc/samba
vi smb.conf
или
vi /etc/samba/smb.conf
```

Теперь можно внести изменения в файл настройки:

```
[myusername]
comment = Public Share
path = /home/mysamba/
#valid users = @users
valid users = @sambashare,fileserver,myusername
#force group = users
create mask = 0770
directory mask = 0770
read only = no
guest ok = yes
browseable = yes
#security = user
wins support = yes
```

Перезагрузитесь или перезапустите *Samba*, чтобы изменения вступили в силу:

```
/etc/init.d/samba restart
```

Выполнив следующие команды, вы сможете создать пользователя и установить для него пароль *Samba*:

```
sudo useradd myusername -m -G users
sudo passwd myusername
```

Введите пароль и затем введите его повторно.

```
sudo smbpasswd -a myusername
New SMB password [Новый пароль Samba]:
Retype new SMB password [Наберите пароль вторично]:
```

Учтите, что *Samba* накрывает всех пользователей в сети. *Samba* позволяет настроить управление пользователями и разрешения доступа только для чтения. После настройки web-сервера (которую мы выполнили выше) вы сможете сразу же протестировать его. Откройте браузер и введите <http://localhost> в адресной строке. Откроется файл */var/www/html/index.html*. Чтобы открыть этот файл на рабочем столе, выберите Home folder [Домашний каталог]. Затем последовательно откройте каталоги *var/www/html*, и вы увидите файл *index.html*.

Щелкните на файле правой кнопкой мыши, и вы сможете открыть его в своем любимом тестовом редакторе. Например, выберите Open with Pluma [Открыть в *Pluma*] или Open With Vim [Открыть в *Vim*]. Теперь можете изменить и сохранить файл. Если по каким-либо причинам сохранить файл не удается, надо проверить, есть ли у вашего пользователя необходимые разрешения для каталога *www/html*. С помощью приведенной ниже команды вы сможете назначить необходимые разрешения для вашего пользователя всем файлам.

```
$ sudo chown -R username:username www
```

Одной из популярных технологий в web-разработке и разработке Android вообще является NodeJS. Чтобы установить и настроить NodeJS, выполните команды

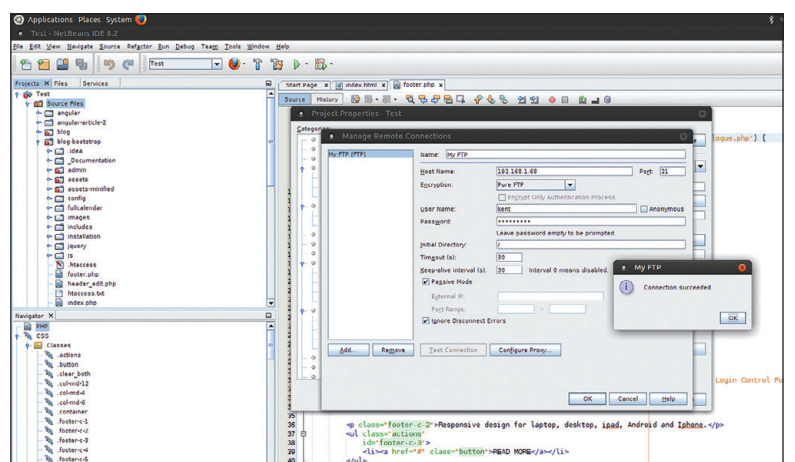
```
sudo apt-get update
sudo apt-get install nodejs
sudo apt-get install npm
sudo apt-get install build-essential
which node
which nodejs
<Вывод>
$ /usr/bin/nodejs
sudo ln -s /usr/bin/nodejs /usr/bin/node
node -v
v4.2.6
```

Итак, NodeJS установлена, и я объясню, как установить *Grunt*, популярный инструмент управления заданиями. Установите *Grunt* и используйте его следующим образом:

- 1 Установите интерфейс командной строки. `sudo npm install -g grunt-cli`
- 2 В командной строке перейдите в каталог, где вы хотите запускать *Grunt* (например, каталог web-приложения или сайта).

Скорая помощь

Скачайте Wordpress с сайта Wordpress.org. Упакуйте файл и переместите полученный каталог в требуемый каталог, например, */var/www/html* в Ubuntu. Желая разместить сайт в корневом домене, скопируйте все файлы в распакованный каталог **Wordpress** и переместите их в каталог */html*.



Успешное подключение по FTP к Raspberry Pi 3 в локальной сети. Raspberry Pi 3 эмулирует удаленный сервер VPS или выделенный сервер.

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

```
cd myfoldername
cd /var/www/html
3 Установите Grunt локально.
npm install grunt --save-dev
4 Создайте простой файл Grunt с именем gruntfile.js и добавьте в него код, приведенный ниже. Этот код позволит вам просматривать сообщения в командной строке при изменении файла test.js. Зачем это может понадобиться? Представим, что вы работаете над этим файлом совместно с коллегой или своими подчиненными. Если файл не изменяется, вы можете разгневаться и спросить, почему никто ничего не делает.
module.exports = function(grunt) {
  grunt.initConfig({
    jshint: {
      files: ['test.js'],
      options: {
        globals: {
          jQuery: true
        }
      }
    }
  });
};
```

```
watch: {
  files: ['<%= jshint.files %>'],
  tasks: ['jshint']
}
});
grunt.loadNpmTasks('grunt-contrib-jshint');
grunt.loadNpmTasks('grunt-contrib-watch');
grunt.registerTask('default', ['jshint']);
console.log('File Changed');
};
5 Создайте простой файл package.json или выполните команду npm init, чтобы создать файл из командной строки. Ниже приведен пример файла package.json со всеми желаемыми и требуемыми зависимостями.
{
  "name": "Test-Project",
  "version": "0.1.0",
  "devDependencies": {
    "grunt": "~0.4.1",
    "grunt-contrib-concat": "~0.1.3",
    "grunt-contrib-jshint": "^1.1.0",
    "grunt-contrib-uglify": "^2.0.0",
    "grunt-contrib-watch": "^1.0.0",
    "grunt-livereload": "^0.1.3"
  }
}
```

Вот как устанавливаются зависимости:

```
npm install grunt-contrib-jshint --save-dev
npm install grunt-contrib-watch --save-dev
npm install grunt-livereload --save-dev
```

Чтобы увидеть *Grunt* в деле, запустите указанную ниже команду, измените файл, а затем посмотрите в консоль, и вы увидите, что файл изменился.

```
grunt watch
```

Gulp

Gulp — это еще одно средство управления заданиями. *Gulp* запускается и настраивается следующим образом (установите *Gulp* глобально, как и *Grunt*):

```
sudo npm install -g gulp
Перейдите в каталог, который хотите использовать. В примере используется каталог gulp, который находится в каталоге html.
cd /var/www/html/gulp
```

Создайте простой файл **package.json** или выполните команду `npm init`, чтобы создать файл из командной строки. Пример файла **package.json** см. ниже:

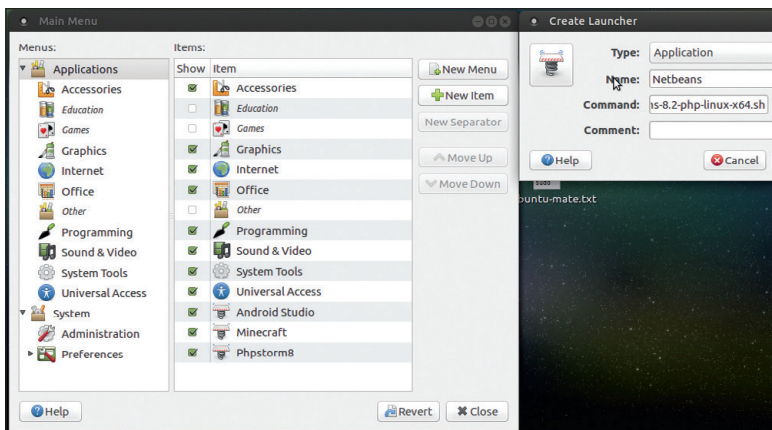
```
{
  "name": "test",
  "version": "0.1.0",
  "devDependencies": {
  }
}
```

Установите *Gulp* локально:

```
npm install gulp --save-dev
```

Создайте файл **gulpfile.js** и добавьте в него необходимые задания. Вот пример файла *Gulp*:

```
var gulp = require('gulp');
livereload = require('gulp-livereload');
gulp.task('watch', function () {
  livereload.listen();
  gulp.watch('*.html').on('change', function(file) {
    livereload.changed(file.path);
  });
});
```



➤ **Добавление Netbeans в главное меню Ubuntu Mate. Добавьте путь и название и нажмите OK.**

Vim: Редактирование в CLI

Vim прекрасно подойдет для доступа к удаленным серверам или для быстрого изменения файлов из командной строки. Кроме того, это великолепный редактор, пригодный для изменения файлов с правами root.

К таким файлам относятся, например, `php.ini`, `/etc/apache2/apache2.conf` и `/etc/php/7.0/apache2/php.ini`.

```
apt-get install vim
```

Если вы не хотите редактировать файлы в *Vim*, можете использовать *Nano*, который входит в состав Linux. Но, уделив *Vim* немного времени, вы увидите, как его удобная навигация помогает экономить время. *Vim* не сохранит файл, пока вы не попросите его об этом. Вот небольшой список команд *Vim*; большая часть команд предваряется нажатием `<Escape>`.

- `G` — переход в конец файла.
- `/stringname` — для поиска строки в файле введите прямой слэш и строку поиска, затем нажмите `Enter`. Для перехода к следующему вхождению нажмите `'N'`.
- `:q!` `<Enter>` — выход из файла без сохранения изменений.
- `:wq!` `<Enter>` — выход из файла с сохранением изменений.
- `:w` `<Enter>` — сохранение файла.
- `<Insert>` — добавление текста.
- `<Escape>` — выход из режима вставки.

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

Установка Netbeans

Netbeans предоставляет великолепные редакторы для всех языков программирования. Есть версии для Java, PHP, HTML/Javascript, C/C++ и не только. Перейдите на сайт <https://netbeans.org> и скачайте Netbeans. Существует множество версий Netbeans, но версия PHP подойдет для web-разработки с использованием стека LAMP. Доступны 32- и 64-разрядная версии. Для работы Netbeans потребуются Java JDK 8. Его можно загрузить и установить из командной строки или с сайта www.oracle.com/technetwork/java/javase/downloads.

Загрузив файлы, щелкните по файлу **netbeans-8.2-php-linux-x64.sh** правой кнопкой мыши, выберите Permissions [Разрешения], разрешите запуск этого файла как исполняемого и закройте окно. Теперь щелкните по этому файлу правой кнопкой мыши и выберите Run [Запустить]. Далее просто следуйте появляющимся инструкциям по установке.

После установки на рабочем столе появится значок Netbeans. При двойном щелчке по нему запустится Netbeans.

Другой способ запустить Netbeans в Ubuntu Mate — выбрать Application > Programming > Netbeans IDE 8.2 [Приложения > Программирование > Среда разработки Netbeans 8.2]. Если вы хотите, чтобы Netbeans отображался в главном меню, выберите System > Preferences > Look and Feel >

Main Menu > New Item > Find Command [Система > Настройки > Внешний вид > Главное меню > Новый элемент > Найти команду], укажите путь, например, **/home/username/netbeans-8.2/bin/netbeans**, и нажмите Close [Заккрыть]. Теперь вы можете получить доступ к Netbeans из главного меню.

Создать проект очень просто. Выберите File > New Project [Файл > Создать проект], выберите HTML/Javascript или PHP, нажмите кнопку Next [Далее], укажите название проекта и нажмите Select Folder [Выбрать папку]. Помните, что по умолчанию используется папка **var/www**, но вам скорее всего понадобится выбрать папку **/var/www/html**. После этого нажмите кнопку Finish [Готово]. Отныне у вас имеется доступ ко всем файлам и папкам в каталоге **www**.

Всё настроено локально, поэтому нам надо настроить FTP для переноса файлов на удаленный сервер. Чтобы настроить FTP или SFTP, щелкните правой кнопкой мыши папку проекта, выберите Properties [Свойства] в разделе Categories [Категории] в столбце слева и выберите Run Configuration [Конфигурация запуска]. Начнется создание конфигурации. Создайте новое подтверждение и укажите его имя. В разделе Run As [Запуск от имени] выберите Remote Web Server [Удаленный web-сервер]. Затем добавьте URL-адрес проекта. После этого выберите Manage [Управление] в разделе Remote

Connection [Удаленное подключение], укажите параметры и нажмите OK.

Проект также можно запустить из удаленных файлов. Вот как это делается. Выберите File > New Project > PHP > PHP Application From Remote Server > Next [Файл > Новый проект > PHP > Приложение PHP с удаленного сервера > Далее], укажите имя проекта, выберите версию PHP, нажмите кнопку Next [Далее], убедитесь, что каталог загрузки существует на удаленном сервере, нажмите кнопку Next и кнопку Finish.

После этого новый проект появится в списке проектов слева. Обратите внимание, что если на удаленном узле нет FTP-сервера, необходимо его установить. Например, это может произойти, если в качестве удаленного узла используется Raspberry Pi. Установить и настроить FTP на удаленном сервере очень легко.

```
sudo apt-get update
```

```
apt-get install vsftpd
```

```
vi /etc/vsftpd.conf
```

Добавьте следующие параметры в файл **/etc/vsftpd.conf**:

```
chroot_local_user=YES
```

```
write_enable=YES
```

```
allow_writeable_chroot=YES
```

Добавив параметры, перезапустите службу: `systemctl restart vsftpd`

```
console.log('File Changed');
gutil.log(gutil.colors.yellow('HTML file changed' + ' (' + file.path + ')'));
});
});
```

Добавьте необходимые зависимости. При этом будет автоматически изменен файл **package.json**.

```
npm install gulp-concat --save-dev
```

```
npm install --save-dev gulp-livereload
```

```
npm install gulp-uglify gulp-rename --save-dev
```

После установки всех зависимостей файл **package.json** будет выглядеть так, хотя он не будет точно таким же, как раньше.

```
{
  "name": "test",
  "version": "0.1.0",
  "devDependencies": {
    "gulp": "^3.9.1",
    "gulp-concat": "^2.6.1",
    "gulp-livereload": "^3.8.1",
```

```
"gulp-rename": "^1.2.2",
"gulp-uglify": "^2.1.2",
"gulp-watch": "^4.3.11"
}
}
```

Чтобы запустить задание 'watch' для отслеживания изменений во всех HTML-файлах, введите в консоли команду

```
gulp watch
```

Для завершения работы *Gulp* нажмите Ctrl+C. Если *Grunt* или *Gulp* не работают правильно, проверьте разрешения для файлов. Если все файлы принадлежат вашему пользователю, всё будет работать как ожидалось и имитировать среду для размещения. Теперь запустим задание для наблюдения за файлами и посмотрим на сообщения в консоли при изменении любого из HTML-файлов в тестовом каталоге. Наряду с мониторингом файлов, *Grunt* и *Gulp* позволяют решать массу других задач.

Дополнения

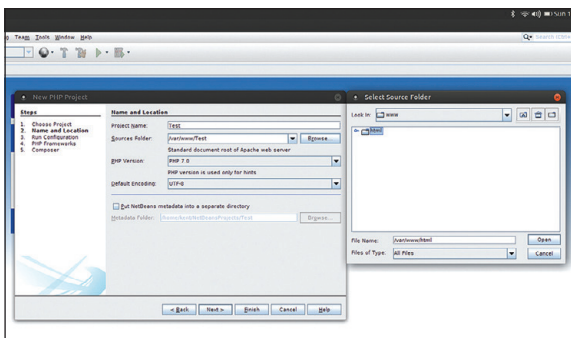
Что касается других программ, то всегда можно установить *Filezilla* для переноса файлов на удаленный сервер. *Filezilla* не только позволяет удобно загружать и скачивать файлы с удаленного сервера из текстовых редакторов, таких как *Vim*, *Gedit* и *Pluma*, но и помогает устранять распространенные проблемы с FTP, так как предоставляет сведения об успешных и завершившихся сбоем операциях передачи файлов.

```
apt-get install filezilla
```

Для запуска *Filezilla* введите команду

```
filezilla
```

Теперь среда web-разработки готова. Вы можете выбрать готовую CMS, например, Wordpress, Joomla или Drupal, и изменить готовые шаблоны, но дальше этого вы вперед не продвинетесь. Чтобы добиться прогресса, в конечном счете потребуются изучать программирование. К счастью, есть множество дешевых онлайн-курсов от Udemy и Coursera. Удачной вам web-разработки. **LXF**



➤ **Запуск проекта Netbeans.** Выберите каталог с исходными файлами, который хотите использовать, и готово.

Python: Движок Шахмат Sunfish

Джонни Бидвелл анализирует нутро небольшого, но ладно скроенного шахматного движка, который обыгрывает его с пугающей регулярностью.



Наш эксперт

Джонни Бидвелл, по слухам, «механический турок» — это объяснило бы металлический скрежет, когда он производит слова в обмен на хлеб и пиво.



Существует легенда о некоем Сиссе ибн Дахире, который изобрел игру в шахматы для индийского короля. Король был так этим поражен, что предложил Сиссе в награду всё, что тот пожелает. Имея склонность к вычислениям, Сисса ответил: «Тогда я хочу, чтобы на первую клетку шахматной доски положили одно зерно пшеницы, на вторую — два, и удваивали количество зерен вплоть до последней клетки: сколько бы ни вышло,

я хочу это получить». Король вскоре понял, что не хватает в мире пшеницы, чтобы выполнить это требование, и опять был поражен. Есть несколько концовок этой истории: в одной Сисса получил должность при королевском дворе, в другой — был казнен за хитрозадость. Надеюсь, наш урок шахмат не приведет ни к казни, ни к погребению Башен LXF под горами пшеницы.

Шахматы — игра сложная: весь «материал» (т.е. фигуры и пешки) ходит по-разному в зависимости от обстоятельств, есть всякие необычные ходы (например, взятие пешки на проходе, рокировка), а пешки получают повышение, если доходят до другой стороны. В результате дилетант в программировании шахмат имеет избыток подводных камней; и вместо того, чтобы тратить целый урок на ловушки, мы позаимствуем код Sunfish у Томаса Але [Thomas Ahle] — полный шахматный движок, написанный на Python. В шахматных движках недостатка нет: от классического GNU Chess до Deep Blue, победившего Каспарова (1997 г.), и навороченного Stockfish. Сами по себе шахматные движки обычно не имеют графического интерфейса; их код в основном направлен на решение немаловажной задачи поиска лучшего хода в данной позиции. Некоторые (включая Sunfish) позволяют играть через текстовую консоль, но большинство обращается ко внешнему графическому интерфейсу, такому как *xboard*, с помощью протокола, подобного Universal Chess Interface (UCI) или WinBoard. Помимо хорошей визуализации игры, это позволяет играть против разных движков из одной программы. Кроме того, мы можем использовать движок против движка и наслаждаться шахматами как зрелищным видом спорта.

Движок Sunfish

Предположим, вы умеете играть в шахматы; а если нет, можете тренироваться, играя против движка Sunfish Томаса. Код вы найдете на **LXF DVD** в каталоге **Tutorials/Chess**. Скопируйте этот каталог в свою папку **home** и запустите его с помощью

```
$ cd ~/Chess
$ python sunfish.py
```

Для отображения в терминале «фишек» (фигур и пешек) программа применяет символы Unicode, благодаря чему больше похоже на шахматы, чем *GNU Chess*. Установите флажок (см. «Установка *Xboard* и взаимодействие с *Sunfish*»), чтобы посмотреть, как получить графическое изображение. Ходы вводятся указанием начальных и конечных координат, поэтому для агрессивного первого хода — перемещения королевской пешки на e4 — надо ввести e2e4. Обратите внимание, что это чуть длиннее, чем более распространенная алгебраическая нотация (в ней предыдущий шаг был бы записан как e4), но значительно упрощает машинам понимание, что вы имеете в виду. Желая сделать рокировку, просто укажите, что хотите переместить своего короля на две клетки вбок: машина знает правила и переместит соответствующую ладью, при условии, что на этом этапе игры рокировка возможна. В зависимости от ваших навыков вы будете побеждать, проигрывать или играть вничью.



Unicode любезно предоставляет значки шахматных фигур, которые улучшают впечатление от игры в терминале.

«Механический турок» и другие шахматные машины

В 1770 г. барон Вольфганг фон Кемпелен [Wolfgang von Kempelen] поразил венский двор «Механическим Турком» — заводным автоматом, сидевшим перед шахматной доской. Кемпелен утверждал, что его изобретение играет в шахматы лучше любого человека. Действительно, барон и «Турок» путешествовали по Европе и поражали зрителей потрясающим талантом последнего.

«Турок» был мистификацией, а его талант на самом деле принадлежал бедняге, прятавшемуся под столом. Тем не менее он вдохновил людей на размышления о шахматных машинах, а в 1950 г. Шеннон и Тьюринг опубликовали статьи на эту

тему. К 1960-м гг. компьютеры уже приемлемо играли в шахматы: Джон Маккарти [John McCarthy] (названный отцом ИИ) и Алан Коток [Alan Kotok] в MIT разработали программу, игравшую лучше большинства начинающих. Эта программа, работавшая на IBM 7090, провела матч по телеграфу против машины М-2 команды Александра Кронрода из ИТЭФ в Москве.

Это был первый в истории матч «машина против машины», и Советы выиграли со счетом 3:1. Их программа превратилась в «Каиссу», в честь богини шахмат, и стала чемпионом по компьютерным шахматам в 1974 г. К началу 1980-х шахматная

общественность задумалась о том, что рано или поздно компьютер победит чемпиона мира. Действительно, в 1988 г. IBM Deep Thought разделила первое место в US Open, хотя на следующий год тогдашний чемпион мира Гарри Каспаров ослепительно его победил. В 1996 г. Deep Blue поразил мир, выиграв свою первую игру против Каспарова, хотя матч чемпион мира выиграл (4:2). Машина была модернизирована, и через год ей удалось победить Каспарова, хотя и не бесспорно. С тех пор компьютеры регулярно бьют эти неполноценные мешки с костями, хотя их мастерство обусловлено достижениями алгоритмов.

В LXF202 мы использовали PyGame для реализации древней настольной игры Gomoku. На этом уроке мы увидим несколько иной подход. Посмотрите на код `sunfish.py`: директива `#!` в первой строке указывает, что `sunfish.py` следует запускать с помощью компилятора Pyru, а не стандартным интерпретатором Python. Установка Pyru проста и значительно улучшит поисковую производительность Sunfish, но для наших целей мы отлично обойдемся без нее. Импортируем синтаксис `print_function` для обратной совместимости с Python 2 и необходимые части других модулей. Затем инициализируем три глобальные переменные, о которых здесь не стоит беспокоиться.

Управление на доске

Теперь приступим к описанию нашей доски. Ее начальное состояние хранится как строка `initial` со 120 символами, что может показаться немного странным, особенно если вы помните, как мило выглядела доска Гомоку в виде двумерного списка. Тем не менее, это представление оказывается гораздо эффективнее. До определения `initial`, примем обозначения угловых клеток, используя стандартную схему: A1 — нижний левый угол, A8 — нижний правый, и т. д. Разделим строку на ряды по 10 символов, памятуя, что

обозначение перевода строки `\n` считается за один символ. Фактически доска начинается в третьем ряду, где мы заменяем черные фигуры стандартными строчными обозначениями, ясности ради перечисленными ниже:

- » р: пешка [англ. pawn]
- » г: ладья [англ. rook]
- » п: конь [англ. knight]
- » б: слон [англ. bishop]
- » q: ферзь [англ. queen]
- » k: король [англ. king]

У нас есть символы, заполняющие начало (пробел) и конец (`\n`) каждой строки, поэтому мы знаем, что перемещение фигуры на одну клетку по вертикали прибавит или вычитет 10 из его индекса в строке. Соответственно, перемещение на одну клетку по горизонтали будет выполняться прибавлением или вычитанием 1, и мы знаем, что если получившийся индекс заканчивается на 0 (т. е. равен 0 по модулю 10) или 9 (т. е. равен 9 по модулю 10), то данное положение находится вне доски. Положение по вертикали 1–8 также можно получать непосредственно в виде второй цифры индекса, а положение по горизонтали — линейно вывести из первой цифры. Пустые промежутки на доске представлены точками (.) во избежание путаницы с пустыми клетками, представленными пробелами.

Используя эту нумерологию, мы описываем отдельные перемещения в направлениях компаса подходяще названными переменными, а затем определяем возможные перемещения каждой фигуры по словарю направлений `directions`. Обратите внимание, что мы здесь определяем перемещения только для белых (т. е. пешки идут на север); их оппоненты могут быть вычислены простой транспозицией. Заметьте также, что мы описываем все направления, в которых фишки способны перемещаться, хотя в текущей позиции какое-то может оказаться недопустимым (например, пешки могут двигаться по диагонали, только когда они что-то берут, а перемещаться на две клетки — только при первом ходе). Мы не учитываем здесь перемещение дальнбойных фигур на две или более клетки по прямой («проскальзывание»), поскольку занимаемся этим в цикле для хода. Затем мы определяем длинный словарь `psf`. В некотором смысле это банк данных движка; он присваивает ценность каждой фигуре в заданной позиции на доске, так что, например, конь (N) скорее будет полезен в центре доски, а вот ферзь — где угодно. Ценность короля настолько непропорционально велика, что машина знает: им нельзя жертвовать никогда.

Теперь перейдем к разделу шахматной логики и подклассу `namedtuple` — это конструкция для описания заданной шахматной

Скорая помощь

В Sunfish был ограничен статический поиск. Это означало, что ходы на пределе глубины не анализировались, что могло привести к так называемым эффектам горизонта, поскольку при этом движок не может видеть прошлые ошибки. Благодаря простой проверке, ходы в этом пределе анализируются, чтобы гарантировать результат в неподвижной позиции.



» Так начинаются все игры в шахматы, но после четырех ходов мы можем оказаться в одной из почти 320 миллионов позиций.

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

позиции. Использование этого типа данных позволяет иметь кортеж (список фиксированной длины) с именованными ключами, а не числовыми индексами. Мы сохраняем текущее состояние board [доски] вместе с оценкой score [счет] этой позиции. Еще у нас есть четыре дополнительных элемента, чтобы позаботиться об особых ходах — рокировке и взятии пешки на проходе. Функция `gen_moves` перебирает все клетки на доске и все возможные ходы для каждой фишки на доске.

Цикл начинается следующим образом:

```
for i, p in enumerate(self.board):
    if not p.isupper(): continue
    for d in directions[p]:
        for j in count(i+d, d):
            q = self.board[j]
            if self.board[j].isspace(): break
```

Функция `enumerate()` (строка 147) является жизненно важным оружием в любом арсенале программиста на Python, поскольку она генерирует пары индекс–значение для данного списка (или строки, в нашем случае); это полезно, когда нас интересуют и положение элементов списка, и их содержание.

В силу симметрии достаточно рассмотреть ходы только белых фишек, поэтому мы выкрутимся, если соответствующая часть `p` в строке не будет прописной (строка 148).

К счастью, метод `.isupper` также возвращает `False` для пробелов и точек, поэтому пустые клетки сразу эффективно отбрасываются. Функция `rotate()` меняет цвета, когда настает ход черных. Мы рассматриваем все возможные направления, в которых фишка может пойти, а затем (строка 150) продлеваем эти перемещения, учитывая, что есть дальнобойные фигуры, способные скользить (т.е. ладьи, слоны и ферзи). Наконец, мы отбрасываем ходы, которые выводят нас с доски.

Следующая часть функции проверяет возможность рокировки:

```
if i == A1 and q == 'K' and self.wc[0]: yield (j, j-2)
if i == H1 and q == 'K' and self.wc[1]: yield (j, j+2)
if q.isupper(): break
```

Право на рокировку для обеих ладей хранятся в виде логических значений в списке `wc`, являющегося частью нашего объекта `Position`. Когда мы рассматриваем любую из угловых клеток белых, и если у белых еще есть право на рокировку (эти клетки действительно заняты ладьями), то мы `yield` [допускаем] ход, перемещающий короля на два пробела влево или вправо. Наш `gen_moves()` — это пример функции-генератора: он дает результаты, пригодные для циклов `for` или `while`. В нашем случае мы создаем пару индексов — положений фишек и после хода. Наконец, мы прерываемся, если целевая клетка занята одной из белых фишек, поскольку свои фишки брать нельзя.

Пешка в игре

Далее мы рассмотрим вопросы, относящиеся к пешкам:

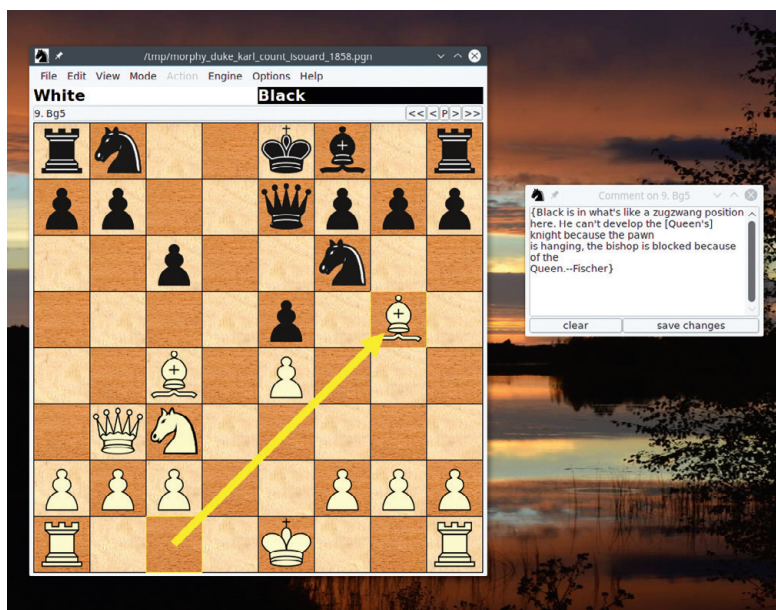
```
if p == 'P' and d in (N+W, N+E) and q == '.' and j not in (self.ep, self.kp): break
if p == 'P' and d in (N, 2*N) and q != '.': break
if p == 'P' and d == 2*N and (i < A1+N or self.board[i+N] != '.'): break
```

Во-первых, пешки не могут двигаться по диагонали на пустую клетку — только при взятии на проходе. Далее, они могут двигаться только вперед (на одну или две клетки, в следующей строке мы проверяем, допускается ли последнее). Сравнение `i < A1 + N` вернет `True` для любой пешки, сошедшей со второй горизонтали. Права хода на две клетки также нет, если перед пешкой что-то стоит.

Заключительные строки `gen_moves()` таковы:

```
yield (i, j)
if p in ('P', 'N', 'K'): break
if q.islower(): break
```

Мы предусмотрели все ограничения и теперь можем перейти к анализируемому ходу; он может оказаться недопустимым (например, если не снимет ситуацию шаха), но прошедшим первый уровень фильтрации. Пешкам, коням и ладьям скользить не разрешается, а другие фигуры обязаны остановить скольжение, когда они что-то берут (т.е. приземляются на запись в нижнем регистре в `board`).



➤ Это из игры Пола Морфи (белые) против герцога Брансуика и графа Исурда в 1858 г. Для черных (их ход) это так называемый цугцванг: большинство ходов губительно.

Установка Xboard и взаимодействие с Sunfish

Для Linux есть ряд хороших графических интерфейсов пользователя; мы используем *Xboard*, так как он довольно распространен в обычных репозиториях дистрибутивов, но обязательно попробуйте *PuChess*. Установка заключается только в

```
$ sudo apt-get install xboard
```

Теперь запустите *Xboard* и выберите Engine > Load New 1st Engine. Введите *Sunfish* в поле *Nickname*, для каталога движка используйте `/home/user/`

Chess (заменив `user` своим именем пользователя — *Xboard*, похоже, не понимает сокращение ~), а для управления движком используйте `python /home/user/Chess/xboard.py`. Оставьте все остальные настройки как есть, и выберите OK.

Настройка *Xboard* по умолчанию предоставляет человеку белые фигуры, и если всё правильно, в заголовке окна теперь должно быть “Sunfish”. Теперь у вас есть сильный противник.

Когда вы нажмете на одну из своих шахматных фигур, *Xboard* великодушно покажет, как ею можно ходить, это очень полезно для начинающих игроков.

Установить *GNU Chess* тоже просто, а *Xboard* уже имеет его в списке Engine. Можно загрузить *GNU Chess* как второй движок и выбрать *Two Machines* в меню *Mode*. Битва должна начаться сразу, и *GNU Chess* даже покажет некоторые из своих безумных мыслительных процессов в строке состояния.

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

Функцию `rotate()` мы уже обсуждали. Но впечатляет, что доску можно перевернуть (что приводит к той же игре при смене цветов и доске, повернутой на 180 градусов), просто взяв строки `board` в обратном порядке и сменив регистр. Однако следует позаботиться и о других составляющих нашего объекта. В частности, следует обратить значение счета, поскольку мы всё-таки находимся в оппозиции к другому игроку, хотя и притворяемся, что приняли его цвет. Правами на рокировку мы занимались отдельно, и они не нуждаются в дальнейшей обработке. Позиции на проходе легко вычисляются обратным отсчетом от конца доски.

Функция `move()` имеет дело с фактическим перемещением фигур, когда приходит время. Сначала мы получаем начальную и конечную позиции `i` и `j` и фигуры на этих клетках `r` и `q`. Строка 178 определяет сокращенную (лямбда) функцию, заменяющей фишку в позиции `i` фишкой, которой мы ходим. Мы получаем и сбрасываем требуемые переменные класса, чтобы лишний раз не набирать `self`, и обновляем счет, вызывая функцию оценки. В строке 184 мы помещаем фишку в новую позицию, применив нашу лямбда-функцию, а затем стираем фишку с ее исходного положения последующим вызовом.

Начиная со строки 187, мы обновляем права на рокировку: если ладья перемещается, то рокировка с этой стороны больше не допускается; значение для другой — неподвижной — ладьи сохраняется. Сама рокировка инициируется королем:

```
if p == 'K':
    wc = (False, False)
    if abs(j-i) == 2:
        kp = (i+j)//2
        board = put(board, A1 if j < i else H1, '.')
        board = put(board, kp, 'R')
```

Как только король перемещен, права на рокировку отменяются, хочет игрок этого или нет. При рокировке король перемещается на две клетки вбок, а ладья — на смежную с королем клетку по другую сторону от короля. Это вычисляется округлением середины между позициями `i` и `j`. Мы используем еще несколько лямбд, чтобы удалить старую позицию ладьи; и последняя строка помещает ладью в новую позицию.

Действия с пешками

Далее мы имеем дело с пешками. `Sunfish` не мелочится, т.е. по достижении 8-й горизонтали пешки всегда обращаются в ферзя (строка 201). Если пешка ходит на две клетки, то мы отслеживаем клетку позади, на случай возможного взятия на проходе. Если пешка берет на проходе, соответствующая клетка стирается. Мы возвращаем новый объект `Position`, не забывая его развернуть с учетом точки зрения другого игрока.

Функция оценки `value()` вычисляет относительную стоимость данного хода. Начнем с вычисления разности между значениями позиций до и после перемещения. Если игрок взял фигуру, то добавляется ценность фигуры в позиции взятия. Если рокировка приводит к шаху королю, то стоимость подпрыгнет до небес (препятствуя этому ходу). Если имела место рокировка, то стоимость следует скорректировать в соответствии с новой позицией ладьи. Наконец, мы должны учитывать превращение пешки и взятие на проходе.

В конце мы дадим обзор раздела логики поиска, но стоит кратко ознакомиться с разделом интерфейса пользователя (который начинается со строки 338). Функция `parse()` преобразует двузначную координатную строку (например, `a4`) в соответствующий индекс списка (в данном случае 61). Функция `gender()` делает обратное. Функция `print_pos()` красиво рисует доску, замещая реальную графику символами `Unicode` и добавляя маркированные оси для вертикалей и горизонталей.

Самая последняя функция, `main()`, устанавливает начальную раскладку и определяет основной игровой цикл. Каждая итерация



» Deep Blue — Каспаров (1997 г., раунд 2). Каспаров сдался, когда машина потрясла его этим ходом, сбив его уверенность на остаток матча. Оказывается, он мог бы вытащить здесь ничью, черт.

начинается с отображения доски и запроса хода. Мы используем регулярное выражение `'([a-h][1-8])'*2` (строка 369), для проверки, что ход правильно оформлен, т.е. как пара координат. Если это так, и ход допустим в текущей позиции (он генерируется `pos.gen_moves()`), то мы продолжаем, иначе — новый запрос. Затем мы разворачиваем доску в сторону компьютера и используем функцию движка `search` для поиска лучшего хода. Если этот ход приводит к мату или мата не удается избежать, то игра закончена; в противном случае ход записывается, и доска обновляется.

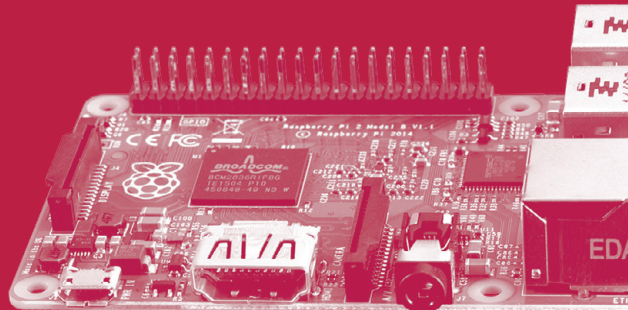
Код, который мы обсуждали до сих пор, легко адаптировать к игре с двумя игроками, как мы видели в учебнике Гомоку. Однако гораздо интереснее код, определяющий следующий ход машины. Удивительно, что сам движок (не учитывая длинный словарь `pst` и весь уже рассмотренный код) занимает менее ста строк.

`Sunfish` основан на минимаксном алгоритме поиска MTD(f), введенном в 1994 г., и адаптированном для использования двоичных деревьев. MTD использует для оценки дерева игры так называемую альфа-бета-обрезку, поэтому мы создаем дерево возможных ходов и отбрасываем все, для которых можем показать, что они приводят к позициям, по-видимому, худшим, чем другие, которые мы оценили. Метод под названием итеративное углубление используется для удержания глубины поиска, чтобы мы не зарывались слишком глубоко в одну конкретную кроличью нору. Вычисление хода начинается с вызова функции `search()`. Мы ограничиваем как глубину поиска (строка 305), так и его широту (используя переменную `NODES_SEARCHED`), чтобы предотвратить выход из-под контроля. Собственно магия происходит в функции `bound()`.

Дерево ходов хранится в упорядоченном словаре `tr`, который индексируется нашей строкой позиций `pos`, и поэтому ранее рассчитанные позиции можно эффективно просматривать. Приступая к анализу всех возможных ходов (строка 270), мы сортируем сгенерированные позиции в обратном порядке по их ценности, что гарантирует подходящим позициям должное внимание. Для построения игрового дерева от каждого возможного хода мы рекурсивно используем `bound()`, добавляя соответствующие ходы в `tr` (строка 293).

Увы, приближается конец статьи, поэтому мы закругляемся.

Вы найдете некоторые полезные замечания, которые помогут вам понять код движка, и почему бы не поэкспериментировать, изменяя параметры и наблюдая, что происходит? Если вы хотите больше узнать о тонкостях программирования шахмат, ознакомьтесь с `wiki Chess Programming` (<https://chessprogramming.wikispaces.com>), это будет ценным ресурсом. **LXF**



Ваша порция смачных новостей, обзоров и учебников от Raspberry Pi

ЛЕС ПАУНДЕР
Будучи нашим постоянным экспертом, он также регулярно пишет о Raspberry Pi.



Привет...

Всего за \$10 можно купить себе целый компьютер — теперь даже с Wi-Fi и Bluetooth. Вы только вдумайтесь! За такие ничтожные деньги можно внедрить в свои проекты Интернет вещей и встроить туда Raspberry Pi. В 2012 г., когда Raspberry Pi только появилась, плата разработки для прототипирования проектов стоила около \$150. Но Raspberry Pi обрушила рынок, уменьшив эту сумму до \$35, и с тех пор Pi стал ассоциироваться с этой стоимостью.

Модель А еще больше снизила цену, но по-настоящему новое падение рынка ощутили только в конце 2015 г., когда вышел Pi Zero. Каких-то \$5 за компьютер — это реально смешно. Мы походя платим столько за кофе. Однако за эти \$5 чего-то всё же не хватало — а именно, беспроводной связи. Можно было использовать USB-модем, но тогда понадобился бы хаб, а хабу — внешнее питание; в итоге у нас был компьютер за \$5, и еще \$15 уходило на аксессуары, чтобы задействовать Wi-Fi, клавиатуру и мышь. Так что Pi Zero, хоть и был революционером по цене, но всё же с оговорками.

Теперь за \$10 мы можем купить Pi Zero W, со встроенным Wi-Fi и Bluetooth 4.1. Но почему это столь важно? Так ведь теперь мне не нужен модем для моего Wi-Fi, и я могу использовать клавиатуру и мышь Bluetooth, а одиночный micro USB-порт OTG отдать для использования с другими устройствами. На самом деле, беспроводная связь Pi Zero W настолько хороша, что я просто подключаю SSH к Pi Zero W и запускаю ее без монитора. Всего за \$10. Вот это сделка!

Raspberry Pi обогнала Commodore 64 по продажам

Теперь Pi — третья по популярности компьютерная платформа на все времена.

Продажи Pi уже превысили более 12,5 миллионов экземпляров, сделав эту платформу третьей по популярности за всю историю. Оставшиеся две — это Apple Mac и IBM PC, в такой компании неплохо быть и третьим. «До недавнего времени Commodore 64 славился тем, что был третьей по популярности платформой для вычислений общего назначения», заявил публике Эбен Урптон [Eben Upton] на вечеринке в честь пятого дня рождения Pi. «Но я здесь, чтобы отпраздновать вот что, — добавил он. — Теперь мы являемся третьей по популярности платформой для вычислений общего назначения после Mac и PC».

Некоторые замечают, что если принимать в расчет весь диапазон Pi, тогда то же нужно сделать и для Commodore 16, 64 и 128; но чтобы превзойти даже эти

общие цифры, потребуется всего лишь еще один год. Так что поздравляем сообщество Pi с преодолением еще одной вехи!



► С64, кое-где используемый до сих пор, был потрясающим домашним компьютером.

Lego Mac Mini

С E-ink и Docker внутри.

Дженнис Херманнс [Jannis Hermanns] создала крошечную копию Macintosh Classic на базе Pi. Внутри него Pi Zero с поддержкой Docker и 2,7-дюймового дисплея с технологией e-ink. Образы Docker реализуются посредством resin.io, а корпус сделан из Lego и спроектирован в приложении Lego Digital Designer. <https://jann.is/lego-macintosh-classic>

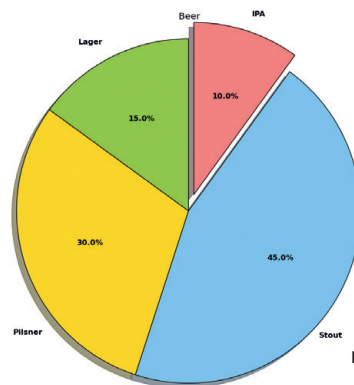


► Такой Mac нам бы пригодился.

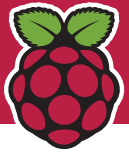
Apt-get Julia

Для любых вычислительных целей.

Язык научного программирования Julia теперь доступен в Raspbian. Исходный код и пакеты уже были доступны в проекте JuliaBerry, но теперь их можно установить в самой системе, используя *Apt*. В JuliaBerry есть Julia-версии API SenseHat, GPIO и MineCraft, доступные на GitHub. <https://juliaberry.github.io>



► Круговые диаграммы в Julia. Вы можете вставить здесь шутку про πr^2 .



Asus Tinker Board

Что это — плата? Какой-то новый Pi? Нет, это еще один претендент на корону Raspberry Pi. Лес Паундер проверяет, достоин ли он титула.

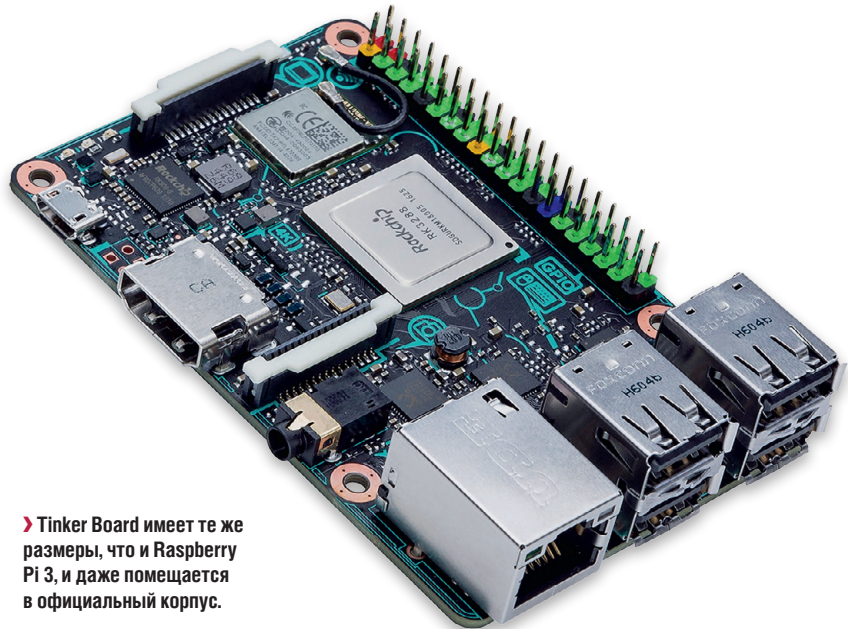
Вкратце

» Плата того же размера, что и Raspberry Pi 3, но с более быстрым процессором, большей памятью и Gigabit Ethernet. Рассчитана на разработчиков, особенно тех, кому требуется больше вычислительной мощности, чем сейчас есть на рынке. Последней тут много, примерно вдвое больше, чем в Pi 3, но нет приличного ПО, чтобы оценить плату от Asus по достоинству.

Плата Asus Tinker Board появилась без лишнего шума, став для многих неожиданностью. Отразилось это в отсутствии программного обеспечения к плате на момент ее выхода: в первые несколько дней в открытом доступе не было даже операционной системы для нее. Но давайте забудем это и приглядимся.

То, что в ее основе лежит система на чипе RK3288 с четырехъядерным процессором ARM Cortex A17, работающим на частоте 1,8 ГГц и 2 ГБ памяти LPDDR3, сразу указывает на то, что данная плата должна была «победить Pi», и в плане вычислительных характеристик так и есть. В тесте на простые числа *sysbench* для одного ядра потребовалось всего две минуты и две секунды, чтобы вычислить все простые числа до 10000. Для сравнения, Pi 3 управился за три минуты две секунды. Выигрыш на добрую минуту! Мы повторили тест, используя все четыре ядра, и Tinker Board завершила его за 31,34 секунды, а Pi 3 — за 45,7. Итак, мы убедились, что мощности в процессоре Tinker Board предостаточно.

Плата оборудована четырьмя портами USB 2.0, а также HDMI, питанием от микро-USB и 40-дюймовым GPIO, не полностью совместимым с платами, произведенными для Raspberry Pi, но способным работать с электронными компонентами (светодиоды, кнопки и т.д.), чтобы создавать авторские проекты. ПО для управления GPIO следует загружать отдельно — почему именно его нельзя было включить сразу готовым, неизвестно. Оно называется ASUS.GPIO, и, как вы уже догадались, это ветка библиотеки RPi.GPIO, которая поддерживает тысячи проектов. Работает оно



» Tinker Board имеет те же размеры, что и Raspberry Pi 3, и даже помещается в официальный корпус.

по тому же принципу, но пока нельзя подключать какие-либо устройства SPI/I2C: для них ПО еще не готово. Сетевые коммуникации представлены встроенным Wi-Fi 802.11 (b/g/n) и Bluetooth 4.0; для приема служит внутренняя антенна, но ее можно заменить на внешнюю для усиления сигнала. Также имеется порт Gigabit Ethernet; однако тест на пропускную способность с использованием *iperf* показал лишь 35,3 Мбит/с. И это всё-таки намного выше, чем у Pi 3, который тянет только 11 Мбит/с, поскольку использует интерфейс USB 2. Tinker Board не объединяет Ethernet-шину с USB-интерфейсом, что обеспечивает более высокую пропускную способность, но не выше реальных скоростей Gigabit.

Asus Tinker Board использует версию дистрибутива Debian под названием TinkerOS. Он легкий и прекрасно работает как настольный, предоставляя доступ к традиционному меню и виджетам для управления беспроводным подключением. Кроме того, вы найдете там web-браузер *Chromium*, и он прекрасно справлялся со всеми задачами, которые мы перед ним поставили — за исключением YouTube. Эта плата способна воспроизводить видео с разрешением 1080 пикселей, но видео на YouTube плохо работает даже после установки заплатки от Asus. Когда мы пытались посмотреть трейлер к *Star Wars Rogue One* [Изгой-один: Звездные войны] с разрешением 1080 пикселей, он зависал и в оконном, и в полноэкранном режимах.

Установив на нашей тестовой машине плейер *Kodi*, мы получили возможность смотреть фильмы HD и передавать HD-контент на наше устройство. *Kodi* работал безупречно, то есть наблюдаемые проблемы связаны только с потоковым контентом через web-браузер, который может быть исправлен последующим обновлением программного обеспечения.

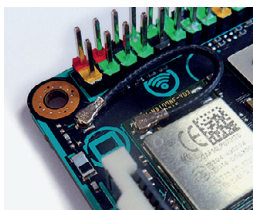
Asus Tinker Board бесспорно является мощнейшей платформой для разработчиков, но, несмотря на всю свою мощь, не способна претендовать на лавры Raspberry Pi, который предлагает больше документации и поддержки тем, кто желает чему-то научиться. Asus Tinker Board — плата для опытных хакеров. **LXF**

Свойства навскидку



Медиа-плейер Kodi

Очень мощный и может отображать видео в качестве 1080 пикселей. На такой небольшой плате.



Внешняя антенна

Wi-Fi-антенну можно отключить, чтобы добавить внешнюю для увеличения дальности действия.

LINUX FORMAT Вердикт

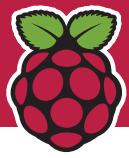
Asus Tinker Board

Разработчик: Asus
Сайт: <http://bit.ly/2oZ6m6Q>
Цена: £55

Функциональность	7/10
Производительность	9/10
Удобство в работе	7/10
Документация	7/10

» Мощная плата с массой возможностей для создания проектов, но ПО и сообществу к ней еще надо расти и расти.

Рейтинг 7/10



iCal: Хитроумный календарь

Проявив смекалку, **Нейт Дрейк** превращает Pi в идеальный ежедневник.



Наш эксперт

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

Скорая помощь

Если вы ранее установили *OwnCloud*, этот проект будет отлично работать со встроенным календарем *OwnCloud*. Просто убедитесь, что все, кому нужен доступ, имеют ссылку LDAP.

Одна из «фишек» таких фильмов, как «Назад в будущее» и «Патруль времени» — в гостиной у каждого есть удобная панель цифрового ежедневника, показывающая назначенные на день дела. Хотя у нас уже есть календари в смартфонах и планшетах, теперь, благодаря Raspberry Pi, вы можете иметь практичный настенный календарь в доме или офисе.

С учетом нашего молчаливого признания, что приложение календаря можно использовать на телефонах, является ли этот проект просто новинкой или у него есть какие-то преимущества? Взглянем правде в глаза. Календарь, установленный на стене, безусловно, может быть эстетичнее, чем многие дисплеи мобильных телефонов. Некоторые люди из Интернета сделали многое в этом направлении — например, со вкусом установили его в деревянной раме или встроили в зеркало.

Главное преимущество, однако, в том, что он может быть доступен другим людям в общественном месте, например, в гостиной. Ваша семья видит ваши назначенные дела, а вы можете быть уверены, что планируете их, сообразуясь с ними.

На рабочем месте для организации встреч и назначения задач коллегам можно использовать такой тип календаря, как «Расписание» в «Google Календаре».

Создание своего календаря

Для этого проекта вам понадобится Raspberry Pi с доступом к Интернету. В интересах экономии на прокладке кабелей и пространстве лучше всего взять Raspberry Pi 3, поскольку в нем есть встроенный Wi-Fi.

Вам также нужно выбрать монитор. Отличным вариантом будет официальный сенсорный дисплей Raspberry Pi (см. «Выбор монитора» на стр. 93), но в конечном счете подойдет любой совместимый монитор. Это не учебник мастера на все руки, поэтому просто попробуйте установить Pi и дисплей, если умеете пользоваться



Совет по оформлению: этот монитор закреплен на деревянной доске и оклеен пробкой, для придания сельского стиля.

дерево и крепить кронштейны. Если дисплей с подставкой, ничто не мешает поместить его на стойку или стол.

Также самое время начать прикидывать длины кабеля, обеспечив, что и монитор, и Pi будут иметь питание, где бы они ни были установлены.

Закончив с оборудованием, выберите тип календаря, который хотите использовать. Если вы и ваша семья или коллеги уже пользуетесь календарем с общим доступом, то можно сразу продолжить действовать по учебнику.

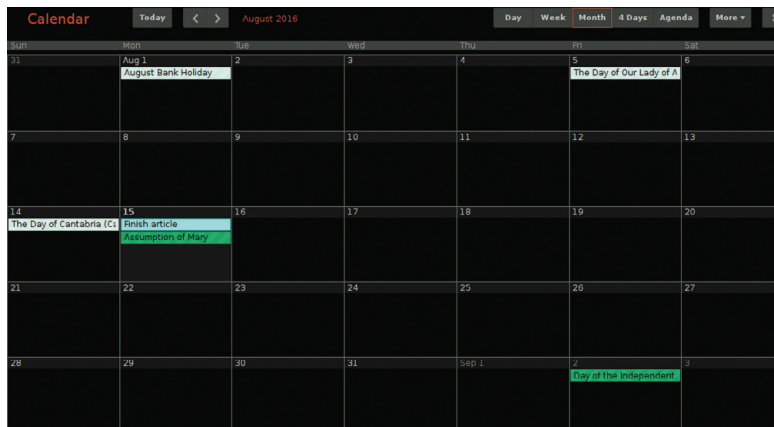
Если это не так, вам может потребоваться создать единый календарь для этой цели. Если вы используете «Календари Google», следуйте инструкциям на странице <http://bit.ly/1HKeoCL>. Пользователи Mac — посетите <http://apple.co/2nz2w1v>, чтобы создать новый «Календарь iCloud». Пользователи Outlook также могут создать календарь, посетив <http://calendar.live.com>.

Не столь важно, какую службу календаря вы используете, лишь бы его можно было отобразить в *Mozilla Firefox*, который мы используем для этого проекта. Попробуйте дать календарю особенное имя, например «Календарь семьи Смитов», чтобы все пользователи знали, что это не их личный календарь.

Импорт календарей

Если у вас уже есть календарь, в новый можно импортировать свои дела, дни рождения и т. д. А может, этого и не требуется, поскольку провайдеры вроде Google и iCloud разрешают использовать несколько календарей. События имеют цветную маркировку, показывающую, какому календарю они принадлежат.

Однако, если один из пользователей вашего нового календаря ранее использовал другую платформу — например, вы решили, что вы все будете использовать «Google Календарь», а один пользовался iCloud на своем iPhone — вам придется его импортировать.



Если вы решите подписаться на другие календари, праздничные дни в других странах и другая полезная информация могут быть выделены.



Выбор монитора

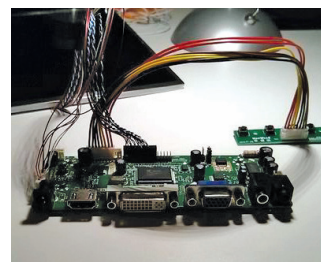
В подходящих маленьких мониторах для подключения к Pi недостатка нет. Если вам не нравится возиться с электромонтажом, официальный 7-дюймовый сенсорный дисплей Raspberry Pi будет идеальным для отображения календаря, он к тому же имеет удобный слот на задней панели для установки Pi. Экран вместе с инструкциями по сборке можно получить на сайте Pi Hut за £55 (<https://thepihut.com>). Если вы не любитель путаться в проводах, Pi Hut также

продает короткий кабель питания микро-USB за £2 для питания Pi от USB-порта монитора.

Сенсорный экран Raspberry Pi имеет дополнительное преимущество, позволяя прокручивать события одним нажатием пальца. Если это вам не важно или дисплей не вписывается в ваш бюджет, Amazon и eBay также продают совместимые с Pi дисплеи. Поскольку Pi имеет порт HDMI, с ним будет работать любой монитор с HDMI, но на некоторых мониторах

есть плата драйвера, позволяющая подключить их к собственному порту DSI Pi.

Если вы в ладах с электроникой и хотите сэкономить, найдите сломанный ноутбук с рабочим ЖК-дисплеем. Если в целости снять экран и купить онлайн совместимую плату контроллера, его можно заставить работать с Pi. Получить некоторые советы можно на www.instructables.com/id/Old-laptop-screen-into-Monitor/?ALLSTEPS.



▶ Плата контроллера магически превращает экран ноутбука в монитор (настройте ее под свой экран).

Чтобы импортировать события из календаря iCloud в Google, сначала экспортируйте их в файл ICS, следуя указаниям на <http://apple.co/110wSQp>. Затем импортируйте файл, выполнив шаг 2 на <http://bit.ly/1mMXSIA>.

Чтобы экспортировать календарь Microsoft Outlook в Календарь Google, выполните действия на странице <http://bit.ly/2cl171N>.

Настройка

Создав единый общий календарь, отведите некоторое время, чтобы выбрать для него удобный формат. Большинство провайдеров предоставляют календари в виде ежедневника, еженедельника или ежесемейника.

Затем можно настроить внешний вид календаря. Изменить календарь iCloud — например, назначить отображаемый период времени — можно по инструкциям на <http://apple.co/2oEuCm2>.

Календарь Google по умолчанию выглядит несколько спартанским. Если вы хотите поэкспериментировать с различными темами, есть несколько доступных на <http://bit.ly/2oeD706>. Вам потребуется расширение Stylish для Firefox, чтобы установить их. Перейдите на <https://mzl.la/1fe2Nwr> и нажмите кнопку Добавить в Firefox, чтобы их установить.

Полноэкранный режим

Поскольку вы будете использовать гораздо меньший экран, чем привыкли, пространство будет на вес золота, поэтому рекомендуем установить дополнение Real Kiosk (r-kiosk) для Mozilla Firefox.

Real Kiosk соответствует своему названию: он превращает браузер в эквивалент интернет-киоска. Это означает, что меню, панели инструментов и даже функция правой кнопки мыши отключены. Главным преимуществом здесь является то, что Firefox всегда открывается в полноэкранном режиме, значительно облегчая просмотр календаря. Это также гарантирует использование устройства только в качестве календаря, поскольку у тех, кто попытается просмотреть другие сайты, ничего не выйдет.

Если вам почему-либо потребуется закрыть Firefox, подключите клавиатуру и, удерживая клавишу Alt, нажмите F4.

Редактирование календарей

До сих пор при чтении этой статьи казалось, что просмотр календаря в web-браузере является пассивным. Однако, если у вас есть общий календарь на стене, не было ли бы идеально позволить людям добавлять и редактировать события?

Если вы решили применить официальный сенсорный дисплей Raspberry Pi, стукните пальцем в любом месте, имитируя там позицию курсора и щелчок левой кнопкой мыши. Этим можно воспользоваться, чтобы редактировать время событий и даже создавать новые.

Проблемы могут возникнуть, если вы захотите редактировать текст событий или дать названия новым. Естественно, можно было бы подключить небольшую беспроводную клавиатуру и оставить ее возле настенного календаря в случае необходимости ввода данных. Однако гораздо менее топорным решением будет встроить клавиатуру в сам браузер. Расширение Mozilla Firefox VKeyboard предназначено для киоск-браузеров и при щелчке на нем открывается, позволяя пользователям вводить текст.

Просто зайдите в браузере на <https://mzl.la/2njGePN> и нажмите Добавить в Firefox для установки. Если вы уже установили дополнение r-kiosk и не можете изменить web-страницу, перезапустите Firefox в безопасном режиме, как описано выше.

Совместный доступ

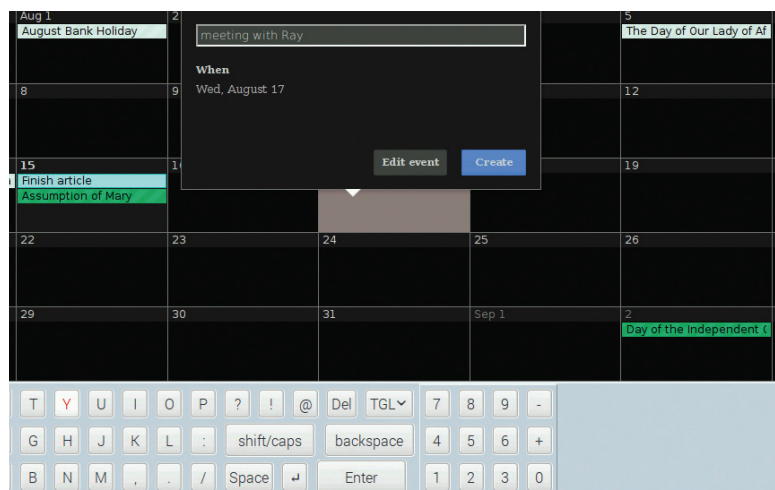
Если вы хотите использовать для добавления или изменения событий в Календаре Google другое устройство, помимо Raspberry Pi, вам надо либо войти в свою учетную запись Google или iCloud на этом устройстве, либо поделиться своим календарем с другими.

Чтобы поделиться своим Календарем Google, следуйте инструкциям на <http://bit.ly/2nzfqwG>. Вы можете отправить ссылку только на конкретные адреса электронной почты или сделать календарь доступным для просмотра всем, у кого есть ссылка. То же самое можно сделать для календарей iCloud, следуя шагам на <http://apple.co/2bfWHk8>.

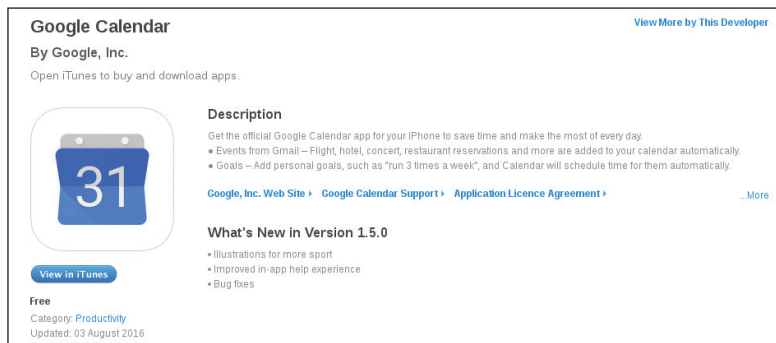
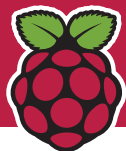
Если вы используете Outlook 2010, также можно опубликовать календарь на Outlook.com, посетив <http://bit.ly/2oCrLzn> и следуя разделу Share a Calendar by Publishing it Online.

Скорая помощь

Если вы уже установили дополнение Real Kiosk, вы не сможете посетить сайт дополнений. Откройте Terminal и запустите `firefox-esr -safe-mode`, чтобы открыть Firefox с отключенными дополнениями.



▶ Используйте плагин VKeyboard для ввода с сенсорного экрана. Клавиатуру можно максимизировать и минимизировать, стукнув по желтой кнопке слева.



➤ Календарь Google позволяет настроить несколько календарей, которые можно отмечать цветом, чтобы отличать их друг от друга.

Когда ваш календарь будет открыт в Сети, тем, кто захочет его редактировать, надо будет иметь к нему доступ со своих устройств. Для тех, у кого есть компьютер, достаточно просто пройти по ссылке в браузере, как на Raspberry Pi.

Также можно просматривать и редактировать календарь на мобильных телефонах. Если общий календарь связан с Google, пользователи Android, даже если у них другая учетная запись Google, могут получить к календарю доступ прямо из своего приложения календаря, следуя инструкциям на <http://bit.ly/2nP8vBB>. Существует официальное приложение Outlook для Android, которое позволяет легко просматривать и редактировать календари Outlook.

К сожалению, с календарями iCloud уже не так легко подружиться, но в магазине Google Play есть несколько сторонних приложений, таких как SmoothSync, которые позволяют синхронизировать календари.

Если вы пользователь iPhone, вам повезло. В iTunes Store есть официальное приложение Google Calendar, через которое можно входить и просматривать свои календари. Есть также официальный Microsoft Outlook — приложение Email and Calendar, пригодное для просмотра и редактирования календарей Outlook.

Головоломки календаря

Если вы создаете или меняете встречу, и она не появляется сразу на всех устройствах, подождите пять-десять минут, прежде чем пытаться устранить неполадки: дайте ей просочиться через различные слои программного обеспечения. Если изменения видны на настенном календаре (то есть на сайте), проблема, скорее всего, связана с устройством, а не с Raspberry Pi.

Программное обеспечение и дополнения, используемые для просмотра календаря, очень просты в установке, поэтому наиболее проблематичной частью этого проекта, видимо, будет подключение монитора и крепление его к стене.

Вы сильно упростите себе жизнь, купив монитор, специально разработанный для Raspberry Pi: тогда у вас будет место, куда поставить сам компьютер — другими словами, аккуратно упрятать его за экраном.

Если место, где вы хотите установить настенный календарь, труднодоступно, можно купить более длинный кабель микро-USB, но помните, что напряжение падает по мере удлинения кабеля. Подумайте о выборе более коротких кабелей и/или USB-концентраторе с питанием.

Если по какой-либо причине в Raspberry Pi произойдет сбой, Firefox попытается восстановить все открытые web-страницы после перезагрузки, и это может означать, что вам надо подключить мышь или клавиатуру, чтобы закрыть все лишние вкладки.

Можно уменьшить вероятность этого, запустив Firefox в безопасном режиме, а затем введя about:config в адресной строке. Нажмите Enter, чтобы перейти к экрану настроек Firefox.

Здесь прокрутите вниз до настройки, обозначенной "Browser.sessionstore.resume_from_crash", и дважды щелкните, чтобы сменить True на False.

Если вы пользуетесь Календарем Google, любой, кто прокрутит в верх экрана, сможет переключиться с вашего календаря на другие ваши приложения Google, такие, как Gmail. Они также могут использовать панель поиска для просмотра документов, хранящихся на вашем Google Диске.

Если вас это беспокоит, рассмотрите вариант создания отдельной учетной записи Google, только для календаря. Вы по-прежнему сможете просматривать и редактировать календарь из своей учетной записи.



➤ Вид сзади на дисплей сенсорного экрана Raspberry Pi, где можно аккуратно разместить Pi. Подставка в комплект не входит.

Скорая помощь

Если у вас назначена личная встреча, вы можете создать закрытое событие. Другие люди, имеющие доступ к вашему календарю, просто видят, что вы заняты, без лишних подробностей.

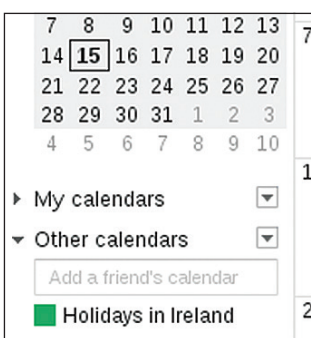
Настройки календаря и советы

Когда календарь пристроен на стену, курсор может выглядеть неряшливо, особенно если вы используете сенсорный экран. Удобное приложение под названием Unclutter может скрыть курсор, когда он не перемещается или вы не касаетесь экрана. Откройте терминал на Pi (или подключитесь через SSH) и выполните команду `sudo apt-get install unclutter`.

На случай сбоя Pi и необходимости перезагрузки, будет лучше запрограммировать Firefox на автоматическое открытие, избавившись от необходимости повторно подключать

клавиатуру и мышь. Откройте терминал на Pi (или подключитесь через SSH) и запустите команду `sudo nano /etc/xdg/lxsession/LXDE-pi/autostart`. Прокрутите в низ окна и добавьте строку `@firefox-esr`. Нажмите Ctrl+X, затем Y, затем Enter, чтобы сохранить изменения.

Наконец, для того, чтобы дисплей поминутно не засыпал, откройте Terminal или подключитесь через SSH еще раз и выполните команду `sudo nano /etc/lightdm/lightdm.conf`. Прокрутите вниз до места, где написано `#xserver-command=X`, и удалите



➤ Google перечислит все календари, на которые вы подписаны.

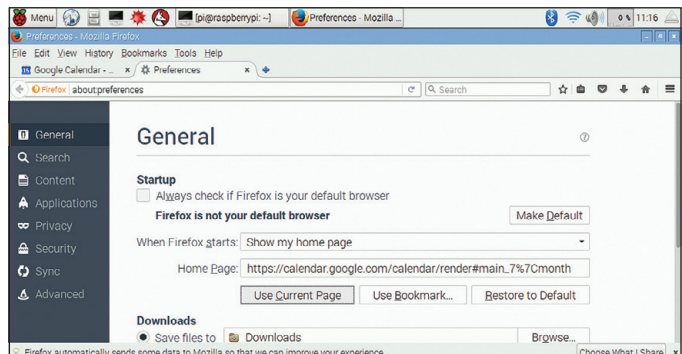
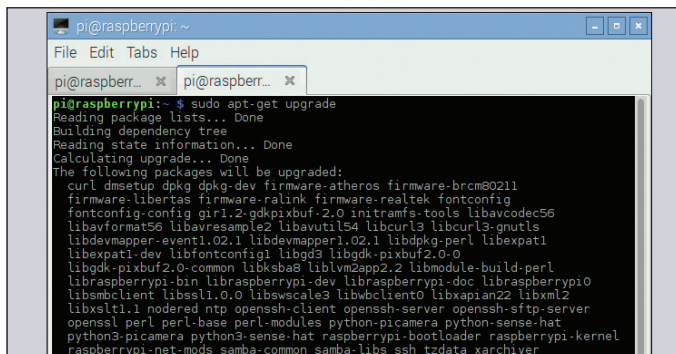
символ # в начале строки. Затем поместите пробел после буквы X и введите `-s 0 -dpms`.

Нажмите Ctrl+X, затем Y, затем Enter, чтобы сохранить изменения, и перезагрузите Raspberry Pi.

Если вы используете Календари Google, щелкните по стрелке рядом с Другие календари, затем по Посмотреть интересные календари, чтобы увидеть список календарей, на которые можно подписаться, например, «Праздники Великобритании». Нажмите Подписать, чтобы они появились в вашем календаре.



Установка настенного календаря

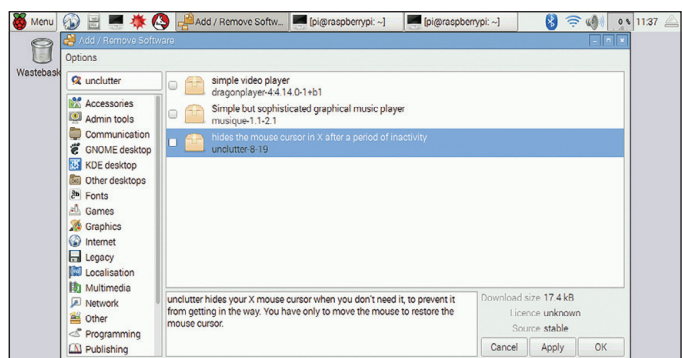
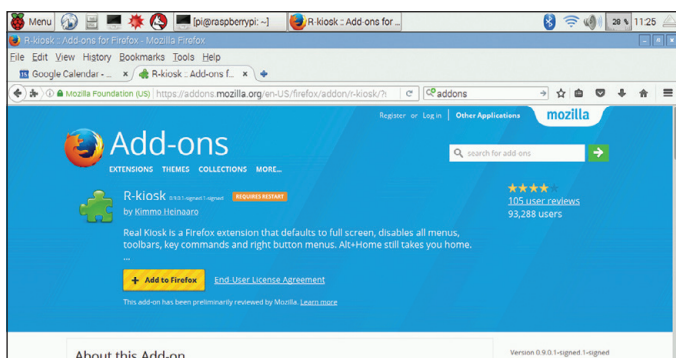


1 Обновите Raspbian и установите Firefox

Прежде чем вы сможете физически превратить Raspberry Pi в настенный календарь, надо подключить устройство к Интернету и открыть приложение *Terminal*. Запустите `sudo apt-get update`, затем `sudo apt-get upgrade`, чтобы обновить Pi. Затем введите `sudo apt-get install iceweasel`, чтобы установить на свой мини-компьютер *Firefox Extended Support Release*.

2 Настройте Firefox

Чтобы открыть *Firefox*, откройте Menu > Internet > Firefox ESR. Посетите свой календарь — <http://calendar.google.com>, например — и при необходимости выполните вход. Если *Firefox* предложит вам вспомнить ваши предпочтения, скажите Yes. Увидев свой календарь, перейдите в Edit > Preferences и нажмите кнопку Set to Current Page, обеспечив, чтобы *Firefox* всегда отображал календарь.

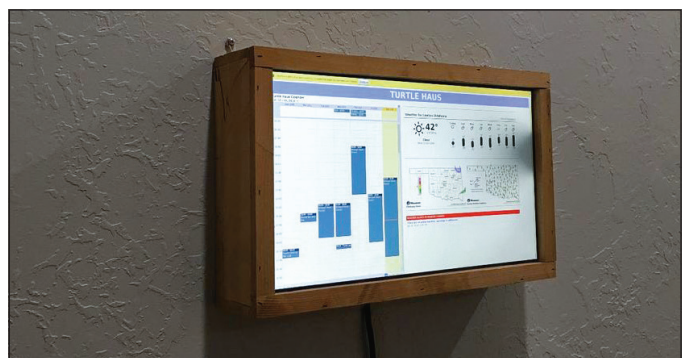


3 Установите Firefox в полный экран

Этот шаг не обязателен, но рекомендуется. Перейдите на <https://addons.mozilla.org/en-US/firefox/addon/r-kiosk> для установки дополнения Real Kiosk — это отключает меню и панели инструментов. Для продолжения работы надо перезапустить *Firefox*. Помните, что вы еще можете закрыть окно, подключив клавиатуру к Pi и нажав Alt+F4.

4 Выполните настройки

Убедитесь, что календарь находится в нужном вам представлении, например, «Ежемесячно». Далее, следуйте инструкциям во врезке «Настройки календаря и советы», чтобы скрыть мышь, когда она не используется, отключить функцию сна Pi и заставить *Firefox* запускаться при каждом включении машины, если хотите. Перезапустите Pi, чтобы изменения вступили в силу.

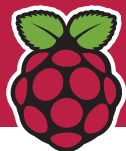


5 Подключитесь

Теперь черед аппаратно-зависимого: подключение всего. Очевидно, что конкретные шаги, необходимые для подключения монитора, будут отличаться в зависимости от устройства. Если вы используете официальный сенсорный дисплей Raspberry Pi, инструкции по сборке можно получить на сайте <http://bit.ly/2nR9qRs>.

6 Заключительные штрихи

Подключив Raspberry Pi к экрану, вы захотите разместить его там, где каждый может получить доступ. Если это в вашем доме, вы, вероятно, также захотите, чтобы он смотрелся как можно лучше. Здесь вы можете позволить своему воображению разгуляться — посмотрите в Интернете, что удумали другие. **LXF**



Minetest: Крафты, ИГРЫ, МОДЫ

Джонни Бидвелл в восторге от того, что теперь есть целых две версии воксельного мира, доступные на Raspberry Pi.



Наш эксперт

Джонни Бидвеллу по душе весь этот крафтинг. Вместо того, чтобы возиться с контейнерами, быстрой разработкой и BUS1, он гораздо охотнее готов выстригивать ложки, сидя в лесу.

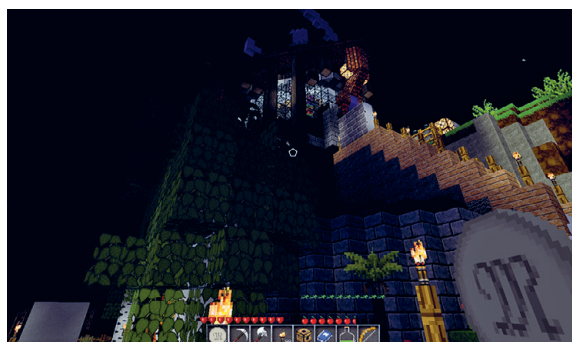
Благодаря API Python, *Minetest Pi Edition* является прекрасным способом знакомства с программированием, а также азами 3D-геометрии. Но эта версия *Minecraft* далеко не такая полная, как коммерческая, и к тому же давно не обновлялась. По этому поводу есть даже петиция к разработчикам *Minecraft*, Mojang (<http://bit.ly/MojangDevMinecraftPi>). Может быть, еще даже не поздно добавить туда свое имя.

Тем временем, у желающих добавить в свои воксельные приложения немного кода, появилась пара новых вариантов. Это дополнения к официальной (платной) версии *Minecraft* (например, *ComputerCraft* в стеке модов Forge), или *Minetest* — клон *Minecraft* с открытым исходным кодом, который можно настроить на собственный вкус при помощи простого в освоении языка Lua. *Minetest* существует с 2012 г., и его последний официальный релиз (0.4.14) вышел в апреле 2016 г. Мы уже рассматривали *Minetest* (см. стр. 54 LXF193); населяющие ее существа и монстры тогда нас впечатлили. Зато не впечатлила ее производительность на Raspberry Pi.

Родной драйвер Pi, VideoCore, использует API OpenGL ES, ориентированный на мобильные приложения; OpenGL ES — это подмножество полнофункционального OpenGL, используемого настольными 3D-приложениями. В движке Irrlicht, на котором работает *Minetest*, использовался экспериментальный рендерер OpenGL ES, но на Pi он не пошел, поэтому обработка снова стала осуществляться средствами ПО, из-за чего играть стало практически невозможно. Однако стараниями Эрика Анхолта [Eric Anholt], у Raspberry Pi теперь имеется свой качественный драйвер OpenGL с открытым исходным кодом, под названием VC4 и с поддерживаемой версии API 2.1. Строго говоря, это драйвер экспериментальный, но при нашем тестировании проблем не возникало. Анholt публикует еженедельный отчет о ходе разработки в своем блоге — <http://anholt.livejournal.com>.

Установить *Minetest* на вашем Pi или даже на вашем Linux-ПК легко (см. врезку «Установка Minetest» вверху стр. 97). Затем вы можете запустить игру из меню Raspbian (или его эквивалента на вашем настольном Linux) или из командной строки, если вам это больше по душе. Мы начнем с самого простого. На вкладке Singleplayer [Режим одного игрока] выберите New world [Новый мир] и дайте ему имя. Прочие настройки оставьте без изменения, и нажмите Create [Создать]. Установите флажок Creative Mode [Творческий режим] и уберите флажок Enable Damage [Включить повреждения]. Это предоставит нам доступ ко всем имеющимся блокам и не позволит навредить самим себе.

Нажав Play [Играть], мы погрузимся в причудливый и постепенно создаваемый мир. Вы можете исследовать его с помощью клавиш WASD, прыгать с помощью пробела и уничтожать/собрать предметы с помощью левой кнопки мыши. После нескольких минут игры наступают сумерки. Вы не можете использовать свой надежный смартфон в качестве дополнительного источника света, как это делают городские жители, когда им приходится провести ночь вдаль от электричества. Но можно пробудить в себе своего



Выглядит этот пафосный балкон с витражной отделкой прекрасно. А вот забраться на него непросто.

внутреннего Макгайвера [Masquuyver — герой американского приключенческого телесериала 1980-х, — прим. пер.] и создать факел. Поскольку мы находимся в творческом режиме, нам незачем беспокоиться о том, как добыть сырье. Все необходимое доступно в меню инвентаря, которое вы можете найти, нажав I. Для изготовления факела нужны палка и кусок угля. Которые, свою очередь, делаются из деревянных досок (любого вида) и угольного блока. Блоки Acacia Planks [Акациевые доски] и Coal Blocks [Угольные блоки] находятся на первой странице вкладки Nodes [Компоненты]; перетащите один из них в слоты инвентаря ниже. Теперь перейдите на вкладку Crafting [Крафт] и поместите Угольный блок в центр. Справа появится девять кусков угля, которые вы также можете перетащить в слот инвентаря.

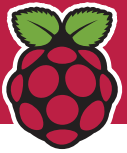
Азбука крафтинга

Точно так же делаются палки из акациевых досок. Чтобы изготовить факел, надо поместить кусок угля по центру, а палку под ним. Слоты инвентаря, в которых хранится несколько единиц материала (например, наши девять кусков угля) можно разделить, перетаскивая правой кнопкой мыши. Половина останется в исходном слоте, а половина появится в целевом. Из одного куска угля и одной палки получится четыре факела. Однако можно создавать объекты из нескольких составляющих и без отходов: если мы разместим в центре все девять наших кусков угля, а все четыре акациевых доски на квадрат ниже, то сможем взять четыре факела с правой стороны, и у нас всё еще останутся те же восемь кусков угля и три палки. Но довольно арифметики, да и в творческом режиме это даже не имеет значения, поскольку всё не ограничено. Итак, давайте вернемся к главному — а именно, к огню! В игре доступен только верхний ряд инвентаря, поэтому поместите туда несколько воспламеняющихся предметов, прежде чем вернуться в игру. Активный слот инвентаря можно изменить с помощью клавиш 1–8 или колесика мыши — выберите тот, где хранятся факелы. Мы не можем носить факелы с собой, это было бы слишком просто; но можем закрепить их на полу (или на стенах, деревьях



Скорая помощь

В *Minetest* можно делать потрясающие вещи. Посмотрите это видео с55 про мод Улучшенных поездов: https://www.youtube.com/watch?v=yFn6VX_8VCM.



Установка Minetest

Последняя версия *Minetest* доступна в стандартном репозитории Raspbian, поэтому для установки достаточно набрать:

```
sudo apt update
sudo apt install minetest
```

А вот чтобы активировать экспериментальный драйвер видео, потребуется чуть больше усилий. Если вы пропустите этот шаг, *Minetest* будет работать, но невыносимо медленно, особенно на одноядерных Raspberry Pi. Начните настройку, используя

```
$ sudo raspi-config
```

Выберите параметр 7 в «Расширенных настройках», затем прокрутите до номера АС (да-да, это тоже номер), GL Driver и поставьте «Да» напротив

драйвера GL. Потребуется перезагрузка, чтобы изменения вступили в силу, что и будет вам предложено при выходе из *raspi-config*; просто выберите «Да». Несмотря на то, что он назван экспериментальным, в наших тестах новый драйвер проявил себя достаточно стабильным, и вряд ли может навредить Raspbian или другим приложениям, которые вы с ним используете. Но если это случится, можно легко вернуться к старому драйверу через *raspi-config*.

Пользователям более ранних моделей Pi с 256 или 512 МБ памяти для работы *Minetest* может также понадобиться увеличить объем, отведенный GPU. Сделать это можно через *raspi-config*, исключая

модели со старой прошивкой, выпущенные ранее октября 2012 г. (если это про вас, читайте о том, как заменить файл **start.elf**). Как мы выяснили, 64 МБ видеопамати достаточно, но более старым моделям с 256 МБ ОЗУ этого может не хватить.

Minetest также доступен в репозиториях большинства дистрибутивов (пользователи Ubuntu для установки могут воспользоваться приведенными выше командами), или же вы можете скомпилировать новейшую версию из <https://github.com/minetest>. О наличии Open GL на настольных системах можно не беспокоиться: сегодня он используется во всех драйверах, как открытых, так и коммерческих.

и даже на потолке) с помощью правой кнопки мыши. Тогда мы можем смотреть на огонь и наслаждаться собственным бессмертием (мы же отменили повреждения, помните?) до утра.

С восходом можем продолжить изучение мира или попробовать свои силы в крафте (т.е. ремеслах). Например, бревен много, и их можно легко преобразить в доски, подобно тому, как мы превратили угольные блоки в куски угля. Рецепты крафтов бывают либо «бесформенными», либо «наделенными формой». В тех, что имеют заданную форму, важно расположение составляющих внутри крафт-сетки, а в бесформенных — нет. Прочитайте руководство по крафтингу на вики <http://wiki.minetest.net/Crafting>, чтобы узнать больше. Помимо создания объектов, есть также плавление, при котором компоненты соединяются в печи. О том, как сделать печь, и о возможностях ее использования вы можете также узнать из вики.

Запуск сервера

Запуск сервера *Minetest* в вашей локальной сети очень прост и позволяет разделить радости многопользовательского режима с вашими друзьями и семьей. Самый простой способ — запустить сервер на том же компьютере, на котором вы играете, хотя мы определенно не рекомендуем делать это на одноядерном Pi. Затем пользователи смогут подключиться к нему с других машин Windows и Linux или других Raspberry Pi в вашей сети. Для этого нужно указать машину, которая будет сервером (лучше, чтобы это была самая мощная машина в сети, но поддержка пары-тройки локальных пользователей вполне по силам современному ПК или Pi 2) и установить *Minetest* на ней и на всех остальных, которые будут ее клиентами. Затем установитесь на сервере и узнайте его локальный IP-адрес. Проще всего — набрав `ip a` в терминале. Затем запустите *Minetest* и перейдите на вкладку Сервер. Создайте новый мир либо используйте существующий, выберите имя для своего игрока (слева) и, по желанию, задайте пароль. Чтобы подключиться к своему серверу или подключить других, запустите *Minetest* на клиентской машине и перейдите на вкладку Клиент. *Minetest* получит из Прекрасного далека список серверов (или, по крайней мере, попытается: иногда возникают проблемы), но не беспокойтесь на этот счет. Просто введите локальный IP-адрес сервера в поле справа, оставьте порт по умолчанию (30000, если вы не изменили его на сервере), выберите имя и нажмите Подключиться.

Как только все присоединятся к игре, игроки смогут бегать туда-сюда, перепрыгивая друг через друга. Вредить друг другу они не могут — это было бы подло. Хотя отправлять сообщения,

чтобы задеть чужие чувства, можно — нажмите `t`, а затем введите «Hello world» и нажмите Enter. Сообщение отобразится всем игрокам. Доступны и другие команды в стиле IRC — чтобы раздобыть их список, введите `/help`. Также можно запустить выделенный сервер (без прикрепленного к нему игрока) прямо из командной строки. Таким образом, сервер *Minetest* можно запускать на машине без монитора или на удаленном VPS. Просто выполните

```
$ minetest --server
```

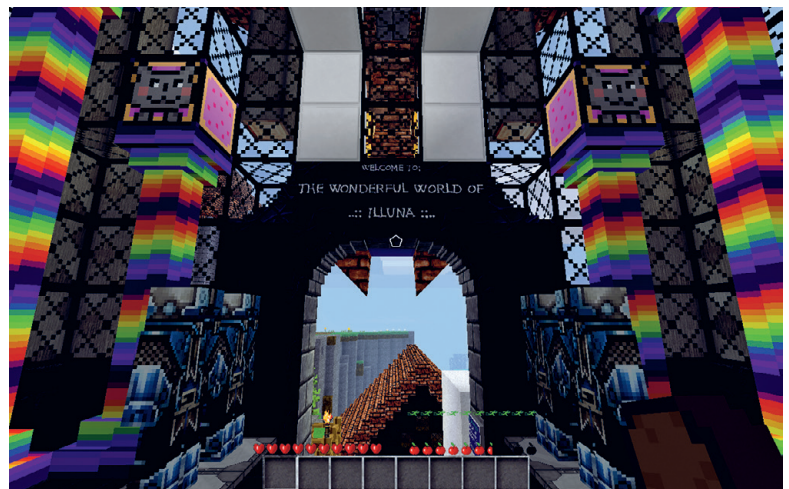
На Raspbian (а также Debian, Ubuntu и их производных) есть минимальный пакет *minetest-server*, у которого нет всех графических зависимостей полного *Minetest*:

```
$ sudo apt-get install minetest-server
```

Затем мы просто выполним `$ minetestserver`, отметив, что это не та команда, которая для полного пакета *minetest*. В этом пакете также поддерживается *Systemd*, так что сервер *Minetest* может запускаться автоматически.

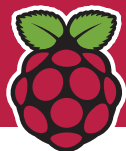
Параметры сервера можно настроить, указав файл с флагом `--config`. Подробную информацию о его формате см. в файле-образце по адресу <http://wiki.minetest.com/wiki/Minetest.conf>. Если вы хотите, чтобы ваш сервер был доступен внешнему миру, надо сообщить серверу TCP-порт 30000 (или тот, что вы укажете в файле настройки либо у параметра `-port`). Файлы создаваемого мира быстро становятся огромными, и мы не рекомендуем размещать

Запустится
и на PC x86!



» *Illuna* — один из самых впечатляющих серверов *Minetest*, которые мы накопили. Узнать больше о его удивительном мире можно на <https://illuna-minetest.tk>.

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



большие объемы данных на Pi. Хранилище SD-карт является узким местом во всех моделях, и сложные конструкции могут съесть всю оперативную память. На солидных серверах *Minetest* мира хранятся в виде базы данных в ОЗУ.

В целом, все настройки игры, моды и миры управляются сервером. Попробуйте зайти на какой-нибудь из многочисленных серверов, чтобы узнать его возможности в плане модов. Нам очень понравилась Illuna и ее Tower of Madness [Башня Безумия], в духе Лавкрафта, статуей кота и грандиозной ратушей (см. <https://illuna-minetest.tk>). Имя для входа на сервер будет использоваться и для создания учетной записи, поэтому пароль стоит запомнить. Изменить его можно в главном игровом меню. Многие серверы не позволяют новым игрокам наносить ущерб или что-либо строить до тех пор, пока они не продемонстрируют активность и адекватность на сервере в течение заданного времени. У каждого сервера своя политика по способам взаимодействия, но если вам припала охота всё бездумно крушить, на это есть много тестовых серверов. TNT-подобные моды довольно часто бывают ограничены из-за чрезмерной загрузки ЦП, которую они могут вызвать.

Моды, игры и миры

Стоит уточнить, где заканчивается движок *Minetest* и начинаются модификации (моды). На вкладке Singleplayer [Режим одного игрока] главного меню вы можете заметить еще два варианта внизу: *Minetest Game* [Игра Minetest] и *Minimal Development Test* [Тест минимального развития]. И то, и другое на языке *Minetest* называется подигрой (или просто «игрой»), работающей поверх основного *Minetest Engine* [Движка Minetest]. Игры — это всего лишь коллекции модов и миров, представляющие собой постепенно создаваемые ландшафты, где игроки строят, обрабатывают, плавают, торгуют или делают что-либо еще, дозволенное им разрешенными модами. *Minetest Game* — это песочница со множеством

текстур и шумов, но без целей или задач на выживание, так что, по сути, это не вполне игра. *Minimal Development Test*, как следует из названия, похож на игру еще меньше (состоит целиком из структур и единственного звукового эффекта) и предназначен только для тестирования.

Minetest Engine, представленный в своей классической форме, еще менее интересен с точки зрения игрока. Он обеспечивает все основные игровые элементы и умения — бег, прыжки, общения и прочее. Но все ремесла, фермерство, голод, текстуры, звуки, погода или еще что-нибудь экзотическое, с чем вы сталкиваетесь во время игры на некоторых из более известных серверов, делаются модами. Много из того, что, казалось бы, должно быть частью основной игры, на самом деле является модом. Это дает некоторое представление о том, насколько настраиваемой является игра — всё ограничивается только пределами вашей фантазии (или, возможно, вашими навыками владения кодом Lua). Например, вы найдете онлайн-игры со знакомыми вам целями — выживание (то есть не умереть от голода и не быть съеденными монстрами), захват флага или создание проблем. В репозиториях Raspbian есть несколько модов. Скажем, вы можете раздобыть больше блоков с помощью

```
$ sudo apt install minetest-mod-moreblocks
```

Также очень легко добавить моды из других источников. Прекрасной отправной точкой является категоризированный список модов по адресу http://wiki.minetest.net/List_of_Mods.

Например, для установки приведенного на этой странице мода *Simple Mobs*, который добавляет крыс, овец и пять страшных монстров (подвижных объектов в мобах мало), перейдите по ссылке на его репозиторий GitHub (<https://github.com/PilzAdam/mobs>) и загрузите главный файл ZIP. Создайте каталог модов и разархивируйте его:

```
$ mkdir -p ~/.minetest/mods
$ cd ~/.minetest/mods
$ unzip ~/Downloads/master.zip
```

Как вариант (если у вас установлен *Git*), можно просто скопировать репозиторий в каталог с модами, используя

```
git clone https://github.com/PilzAdam/mobs.git
```

Теперь создайте в игре новый мир и нажмите Настроить. Список модов появится справа — все те, что требуются по умолчанию, и наши недавно загруженные моды. Выделите моды и установите флажок Включить. Нажмите Сохранить, а затем Играть. Теперь вы видите овец, которые бродят по склону холма, а ночью на вас могут напасть враждебные орки.

Для разработки модов *Minetest* использует скриптовый язык Lua. Некоторые из наших читателей уже ознакомились с API Python от *Minecraft* и немного умеют работать с Lua. По словам Перту Ахолы [Perttu Aholia] (он же celeron55/c55), разработчика оригинала *Minetest*, «Lua — самый простой в мире быстрый скриптовый движок для интеграции с встроенными программами, и он имеет свободную лицензию. Как только вас перестанет бесить синтаксис и вы добавите необходимые вам инструменты, Lua становится

➤ Эти грибы, похоже, из семейства мухоморовых, так что есть их не стоит. Ни в игре, ни в жизни.



API Minetest

В *Minetest* существует три основных класса объектов:

- **Компоненты** Блоки, из которых состоит мир.
- **Инструменты** Оружие или другие орудия.
- **Материалы** Различные предметы (например, палки, бумага, минералы).

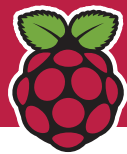
Когда вы дойдете до создания собственных модов, все новые объекты вам надо будет указать в файле с именем *init.lua*, используя, например,

метод `minetest.register_node()`. Каждый мод должен содержать по крайней мере этот файл инициализации, либо внутри директории `~/.minetest/mods` — для пользовательских модов, либо `/usr/share/minetest/mods` — для глобальных.

Помимо объектов, вы также можете создавать рецепты крафтов. Добавив следующее условие в файл *init.lua* мода, мы сможем создать радужный блок Nyuan Cat с помощью яблока и блока грязи:

```
minetest.register_craft({
    output = "default:nyancat_rainbow" 1,
    recipe = {
        {default:dirt, 'default:apple', ""},
        {"", ""},
        {"", ""},
        {"", ""},
    }
})
```

Вот бы так в реальной жизни...



Я не всегда строю гигантские башни, но когда строю — ставлю рядом гигантский кран, хоть он и не имеет отношения к самой постройке.

похожим на JavaScript, но более минималистским». (Очень краткое введение см. во врезке *API Minetest*, внизу стр. 98).

Знакомьтесь: м-р Minetest

Нам удалось встретиться с Пертту Ахолой, который рассказал нам немного о том, как начинался проект: «Я открыл для себя *Minecraft*, когда он был еще на стадии Alpha (в октябре 2010 г.). Тогда не было еще даже многопользовательского режима выживания, и я стал спрашивать некоторых своих друзей из IRC: а насколько трудно сделать такую штуку, и как это делается? Через несколько дней я придумал первую версию *Minetest* — у меня ее больше нет, но если кто-то захочет увидеть одну из ранних версий, загляните на <http://packages.8dromeda.net/minetest/minetest-c55-win32-101010000530.zip>. Это сборка для Windows — мне нравится выпускать все в виде пакетов, совместимых с *Wine*, поскольку это позволяет охватить максимальную аудиторию всего одной сборкой.

Моих познаний в C++ тогда было достаточно, чтобы создать кое-какие любительские проекты GUI, используя wxWidgets и т.д. Сейчас, оглядываясь назад, я понимаю, что с C++ у меня было слабовато: я еле-еле понимал, что такое исключения... У меня также не было опыта работы с 3D-графикой, я только слегка разбирался

в 2D SDL, поэтому мне нужно было нечто [движок] позволяющее быстро создавать индивидуальный контент.

Выбор, по сути, сводился к Ogre или Irrlicht. Основная проблема была в лицензировании, и я выбрал Irrlicht, поскольку Ogre тогда выходил под LGPL. На том этапе я не был уверен, что хочу взяться за создание проекта FOSS. Я просто научился тому, что мне было нужно, переделывая примеры пользовательской геометрии Irrlicht. Мне также требовалась многопоточность, и я выяснил, что здесь мне подойдет JThread. Не помню, чтобы раньше я делал что-то связанное с многопоточностью, просто мне подумалось: „Интересно, насколько это сложно?“ [Смеется.]

Irrlicht, на самом деле, штука несколько устаревшая — честно говоря, так было даже в 2010 г.; но он работает. И в этом вся его прелесть: в нашем мире слишком много вещей, которые на самом деле сопротивляются, когда вы пытаетесь их употребить. Итак, я использовал *Vim*, C++, Irrlicht и JThread около двух месяцев. Затем добавил *zlib* для сжатия данных Миров и *Stake*, чтобы сделать удобнее процесс создания объектов. Это были основные компоненты *Minetest* примерно года два — только самые базовые инструменты и несколько надежных библиотек — вот всё реально необходимое, чтобы что-то создать. Ну и немало решимости. Когда сформировалось сообщество и люди начали присылать заплатки, я решил сделать *Minetest* проектом FOSS, и теперь всё выпущено под LGPL.»

В этом году Ахола начал работать с Raspberry Pi. Посмотрите удивительный мод автомобильного стерео в его блоге по адресу <http://c55.me/blog>. Правда, финская зима для него слишком сурова: «Чтобы запуститься при минус 20, ему следует немного прогреться. К счастью, регуляторы на стереоусилителе очень мощные, и способны поднять в системе температуру выше минусовых отметок». Помимо нелюбви к холоду, у него есть кое-какие претензии и к Pi: «У Pi недостаточно флэш-памяти для ОС, а SD-карты недолговечны, что делает Pi, по большей части, игрушкой. Кроме того, интерфейс GPU по-прежнему является проприетарным». Однако это не помешало Ахолое выпустить новую версию *Minetest*. Версия 0.4.15 включает дрезины для езды по рельсам, улучшенный генератор карт V7 и создание эффекта сферического тумана. Наслаждайтесь! LXF



Это не просто реквизит: на этих консолях вы можете сыграть в настоящий, хоть и немного корявый *Тетрис* на базе Lua.

Скорая помощь

Один из наиболее продвинутых модов называется *Mesecons*. В нем можно создавать электрические переключатели, сенсоры, поршни и даже цифровые замки. Подробнее на mesecons.net.



LXFEHotPicks



Александр Толстой

Mate » MtPaint » Meteo-Qt » NTFS-3G » Guetzli » LanguageTool » Webenginepart » Man vs Olives » Tank Island » GNU Nano » Classifier

Подает к столу большую кастрюлю горячего супа с открытым кодом, щедро приправленного полной свободой, которого вам должно хватить, чтобы продержаться еще месяц.

Среда рабочего стола

Mate

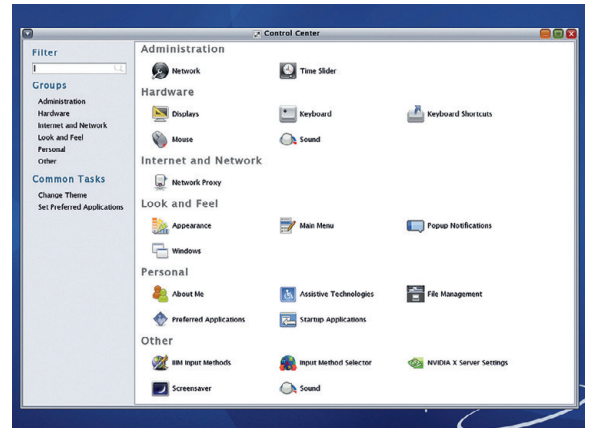
Версия: 1.18 Сайт: <https://mate-desktop.org>

Мир рабочего стола Linux является предметом горячих обсуждений и ожесточенных онлайн-схваток уже не первый десяток лет; но есть одна вещь, которая с годами вообще не меняется. Естественно, мы подразумеваем Mate, преемника некогда знаменитого и широко используемого рабочего стола Gnome 2.

Оглянувшись на несколько лет назад, мы увидим Gnome 2 в качестве стандартного рабочего стола для первых релизов Fedora Core, Novell Enterprise Linux, OpenSolaris и многих других, с уклоном в сторону основанных на Unix рабочих столов уровня предприятия. Любопытно, что

Mate выглядит и работает в точности как Gnome 2, предлагая приятные обновления и поддержку, которые выводят его на один уровень с другими современными рабочими столами.

Возможно, он не настолько требователен, но не менее функционален, и является хорошим выбором для тех, кому надо, чтобы их компьютер работал, а не выдавал интересные эффекты рабочего стола,



» Mate имеет центр управления, где собрались все примечательные настройки и строят против вас заговор.

Mate избавился от уже старой (но используемой) библиотеки GTK2.

но при этом ухитрился не выглядеть староватым и урезанным, как LXDE.

Mate 1.18 вышел спустя почти полгода усердного труда после версии 1.16, и в нем довольно много достойных внимания изменений. Наконец-то в этом релизе разработчики Mate избавились от уже старой (но по-прежнему широко используемой) библиотеки GTK2 и полностью перешли на GTK3. Это означает не только совместимость с самыми последними интересными темами GTK, но и продвижение более легкого в поддержке кода.

Менеджер файлов Caja предлагает копию очереди и паузу, а также уведомления для безопасного размонтирования внешних дисков. Просмотрщик документов Atril обрабатывает PDF намного бойчее обычного, калькулятор Mate снова предлагается по умолчанию, а менеджер архивов Engrampa теперь открывает ZIP-архивы, которые делают вид, что являются кастомными форматами, как WAR и EAR.

Этот новый релиз Mate, естественно, предлагает и серию мелких изменений, и вам определенно надо начать использовать этот рабочий стол, чтобы извлечь выгоду от всех сделанных недавно улучшений. Многие новые функции предполагают интенсивное использование дополнительных кнопок мыши и колеса прокрутки, и уж будьте уверены: разработчики Mate действительно позаботились о продуктивности вашей работы.

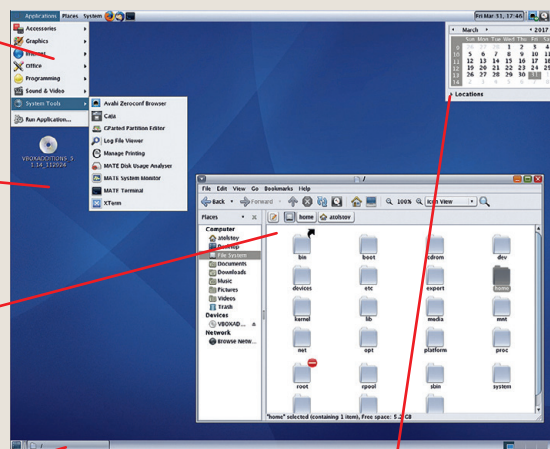
Исследуем интерфейс Mate

Меню приложений
MATE предлагает классическое меню приложений с иерархической структурой.

Рабочий стол со значками
Дом для некоторых значков по умолчанию и место, куда вы можете скинуть свои файлы.

Менеджер файлов
Caja — это то, что много лет назад было Nautilus. В нем нет недостатка в жизненно важных функциях!

Нижняя панель
Эта вторая панель используется для переключения между задачами и виртуальными рабочими столами.



Системный лоток
Календарь, громкость и уведомления работающих приложений размещаются здесь.

Программа рисования

MTPaint

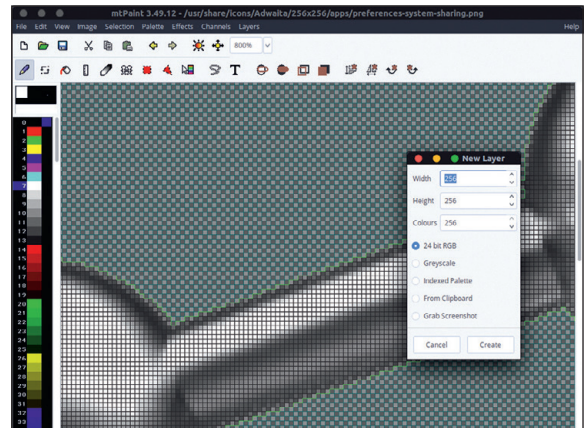
Версия: 3.50 Сайт: <https://github.com/wjaguar>

У нас имеется еще одно дополнение к вашей подборке исключительно легковесных рабочих приложений. В прошлый раз (см. LXF223) мы восхищались отличной программой рисования *AzPainter*, и если вы сочли ее интерфейс слишком запутанным, то вот вам кандидат попроще. Встречайте *MTPaint*, персональный графический редактор, написанный Марком Тайлером [Mark Tyler] еще во времена *GTK1* и изначально оптимизированный для достойной производительности на компьютерах в сто раз слабее вашего нынешнего смартфона.

Однако речь не идет о старом заброшенном оборудовании. *MTPaint* постоянно развивался до версии 3.40 (вышедшей в 2011 г.), но с того времени вышли только пара-тройка версий для разработчиков. Текущий релиз *MTPaint* 3.50 предлагает впечатляющее число улучшений и новых функций, и, кстати, его можно спокойно скомпилировать с *GTK2*. Итак, теперь вы можете наслаждаться консолью скриптов

(Image > Script [Образ > Скрипт]), поддержкой нескольких потоков при рендеринге изображений (хорошо для многоядерных CPU), опциональным исправлением гаммы для живописи, намного лучшими текстовыми инструментами, многочисленными улучшениями инструмента клонирования, новым форматом файлов (PMM) и многим другим. При этом *MTPaint* остается классическим графическим приложением, оптимизированным для ручного рисования и работы с индексированными палитрами. Ему требуется столько же ресурсов, сколько, скажем, *Microsoft Paint* из ранних версий Windows, но взамен он дает вам более расширенные инструменты. *MTPaint* поддерживает слои, прозрачность, выделение, до 1000 шагов отмены и до 8000%

Для многих обычных задач MTPaint вполне может заменить GIMP.



► Превратите старый компьютер в мощную графическую станцию с помощью *MTPaint*!

масштабирования [Ред.: — Чтобы разглядеть повышение наших зарплат]. И даже этим не ограничивается список функций *MTPaint*: как насчет экспорта вашего рисунка в ASCII или превращения многослойного файла в GIF-анимацию всего несколькими щелчками мыши? В порядке альтернативы, можете настроить собственные индивидуальные сочетания клавиш клавиатуры для любого пункта меню редактора и заделаться растровым клавиатурным ниндзя.

Для многих обычных задач *MTPaint* вполне может заменить *GIMP* как основной инструмент творчества, так почему бы не дать шанс этому маленькому, но мощному редактору?

Приложение прогноза погоды

Meteo-Qt

Версия: 0.9.5 Сайт: <http://bit.ly/2ppd0Am>

Вот уже не первый год мы собираем удобные приложения рабочего стола, способные помочь вам создать индивидуальный и легковесный рабочий стол, который будет делать именно то, что вам надо, без полчищ сервисов и движков, сжирающих ресурсы вашего CPU. Отличное *Meteo-Qt* как раз относится к таким приложениям, и вряд ли ему нужно дальнейшее представление. Однако приложений погоды несметное количество, и вот почему вам стоит выбрать именно это.

Meteo-Qt — компактное приложение, которое восседает в системном лотке и показывает подробный прогноз погоды по нажатию на его значок. Текущие данные, как и прогноз, скачиваются с сайта OpenWeatherMap, так что вы узнаете температуру, скорость ветра, облачность, давление и влажность вместе с временем восхода и захода почти для любой точки планеты. *Meteo-Qt5* использует компоненты Python и *Qt5*, но, в отличие от разных плазмойдов

QML с такими же функциями, не использует *QML* и поэтому отлично подходит для любого рабочего стола. Чтобы *Meteo-Qt* функционировал на вашем рабочем столе, проверьте наличие у вас привязок Python 3 к *Qt5*, *LXML* и *SIP* (установите их пакеты через свой менеджер пакетов). Когда все упомянутые зависимости времени исполнения будут удовлетворены, приложение можно запустить из исходников командой

```
$ python3 /path/to/meteo-qt/meteo_qt/
meteo_qt.py
```

Когда приложение загрузится, щелкните правой кнопкой по его значку в системном лотке и перейдите в раздел Settings [Настройки]. Здесь можно изменить свое местоположение, блоки, параметры

Не использует QML, и подходит для любого рабочего стола.



► *Meteo-Qt* дает вам полный прогноз погоды.

подключения, вид значка в лотке, а также его работу, размер шрифта, режим автозапуска и прочие полезные настройки. *Meteo-Qt* требует, чтобы вы зарегистрировались на OpenWeatherMap и получили личный ключ для привязки части рабочего стола *Meteo-Qt* к вашей учетной записи онлайн. Это займет пару минут, и с тех пор не будет беспокоить вас вообще. Значок обновляет прогноз погоды каждые 30 минут, что опять же можно изменить по вашему желанию.

Драйвер файловой системы

NTFS-3G

Версия: 2017.3.23 Сайт: <http://bit.ly/2o3rp0q>

Несмотря даже на то, что рыночная доля Microsoft Windows (очень) медленно уменьшается, мы по-прежнему используем разные ухищрения для облегчения взаимодействия Linux и Windows. Tuxera — успешный поставщик драйверов файловых систем для многих платформ, флагманом которого является широко известный NTFS-3G.

Хотя компания зарабатывает на продаже коммерческих программ, NTFS-3G является технологией с полностью открытым кодом. Как можно понять по названию, драйвер активирует работу с разделами NTFS на любой платформе не-Windows, способной работать с файловыми системами уровня пользователя, также известными как FUSE (`$ sudo modprobe fuse`). Помимо Linux, список включает множество других ОС, от macOS до Haiku и OpenIndiana. Пакет NTFS-3G содержит базовую часть для монтирования и пакет *ntfsprogs* для действий с разделами. Вы можете читать, писать и изменять размеры разделов NTFS без

потери данных, и поскольку драйвер уже некоторое время считается готовым для предприятия, использовать NTFS-3G совершенно безопасно. Чтобы монтировать раздел NTFS, используйте следующий шаблон:

```
$ mount -t ntfs3 /dev/sdb1 /mnt/Windows
```

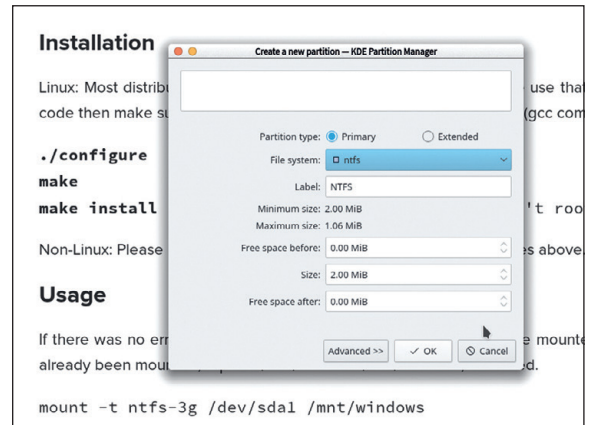
Сканирование и исправление возможных ошибок проходит так:

```
$ ntfsfix /dev/sdbX
```

Не забудьте изменить *sdbX* на реальное имя своего раздела NTFS.

В этом новом релизе имеется несколько улучшений — например, способность монтировать тома NTFS в режиме только для чтения в случае, если Windows поместила их в так называемое состояние «сна», усовершенствование работы с расширенными атрибутами, улучшенная поддержка UTF16

Помимо Linux, список включает множество других ОС.



► Безопасный и надежный способ работы с разделами Windows из Linux.

и многое другое. У драйвера в целом имеется несколько отладок, оптимизирующих чтение и запись из тома NTFS из Linux любой FUSE-совместимой ОС.

Версия 2017.3.23 является первым основным обновлением Tuxera NTFS-3G почти за год, так что его стоит обновить, если вам нужна поддержка файловой системы NTFS под Linux — например, если у вас имеется Windows на отдельном разделе или вам надо работать с внешними дисками, отформатированными с помощью NTFS.

Кодировщик изображений

Guetzli

Версия: GIT Сайт: <https://github.com/google/guetzli>

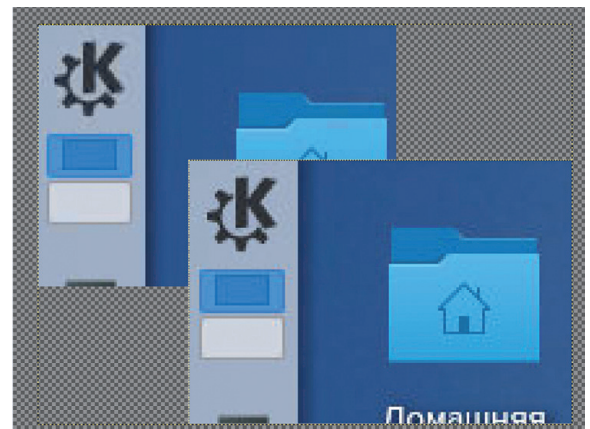
Мы гордо продолжаем наше путешествие в мир альтернативных кодировщиков медиа. Мы уже рассказывали о *FLIF* (LXF205) и *Lepton* (LXF215) для выжимания всех лишних битов из графики, и об *Opus* (LXF209) для отличной работы с радио. *Guetzli* относится к лагерю графики — это новый кодировщик JPEG от Google, целью которого является дальнейшее уменьшение размера файлов JPEG по сравнению с *libjpeg*, библиотекой, которую можно найти почти в любом дистрибутиве Linux. Google сообщает, что *Guetzli* в состоянии сэкономить до 30% места на диске без ущерба качеству (хотя JPEG уже является форматом с потерями). Мы решили испытать *Guetzli* и самолично увидеть, стоит ли включать его в нашу реполненную директорию `/usr/bin`.

Превращение скриншота PNG в JPEG показало, что у *Guetzli* с настройками по умолчанию практически нет преимуществ перед *libjpeg* по размеру. По умолчанию *Guetzli*

использует 95% качества, но допускается уменьшать его до 84% — минимально возможного показателя для *Guetzli*. Этот коэффициент сжатия очень хорош для JPEG, так что мы использовали его и, наконец, получили обещанные цифры, 260 КБ файл против 206 КБ в нашем случае.

Guetzli имеет и некоторые другие ограничения: он производит только непоследовательные JPEG и требует немало системных ресурсов. Инструмент командной строки поддерживает ограничение на отведение памяти, с огромным значением 6000 МБ по умолчанию. У нас ушло около пары минут (!) на превращение вышеупомянутого скриншота в JPEG с помощью *Guetzli*, хотя другие библиотеки, например,

Отличный инструмент для разовой оптимизации библиотеки.



► *Guetzli* (наверху) производит меньше артефактов в меньшем по размеру файле.

libjpeg или *MozJpeg*, намного быстрее. Однако *Guetzli* был создан, чтобы сэкономить вам немного места на диске.

Маловероятно, что вы решите использовать *Guetzli* для кодирования на лету, но это отличный инструмент для разовой оптимизации вашей библиотеки изображений. *Guetzli* тщательно сохраняет совместимость с любыми сторонними кодировщиками JPEG, и вы должны быть в состоянии просматривать свои файлы так же, как вы это делаете сейчас. Единственный ресурс, который вам потребуется, это время и лошадиные силы CPU (бойтесь счёта за электричество!).

Программа проверки орфографии

LanguageTool

Версия: 3.7 Сайт: <https://languagetool.org>

При переходе с Windows на Linux одним из наиболее востребованных видов приложений является офисный пакет. Честно говоря, их полно — как с открытым кодом, так и проприетарных. Но дело не ограничивается должным анализом в Linux файлов в форматах DOCX и XLSX, потому что обычные пользователи Windows рассчитывают на качество проверки орфографии, привычное им в *Microsoft Word*, который привлекает некоторые сторонние проприетарные языковые инструменты.

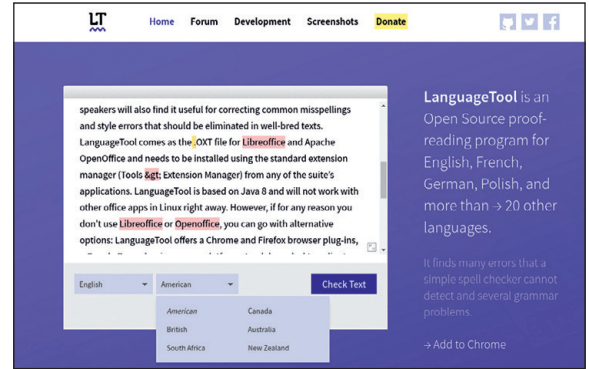
Хотя стандартная проверка орфографии в таких пакетах, как *LibreOffice*, работает очень хорошо благодаря библиотеке *Hunspell*, мы могли бы пожелать большего: в частности, автоматической проверки грамматики, пунктуации и стиля.

LanguageTool — это расширение, которое реализует все вышеперечисленные возможности более чем для 20 языков, по большей части европейских. Те, для кого родным является английский, тоже найдут

его удобным для исправления обычных опечаток и стилистических ошибок.

LanguageTool поставляется как файл OXT для *LibreOffice* и *Apache OpenOffice*, и его следует устанавливать через стандартный менеджер расширений (Tools > Extension Manager) из любого приложения пакета. *LanguageTool* основан на Java 8 и не будет сразу работать с другими офисными приложениями в Linux. Но если в силу каких-то причин вы не используете *LibreOffice* или *OpenOffice*, можете использовать альтернативные опции: *LanguageTool* предлагает плагины для браузеров *Chrome* и *Firefox*, плагин Google Docs, кросс-платформенный клиент рабочего стола и, наконец, поле ввода текста онлайн на сайте проекта. При использовании в офисном пакете

Дает автоматическую проверку грамматики, пунктуации и стиля.



Для проверки текста с помощью *LanguageTool* не обязательно использовать *LibreOffice*.

LanguageTool заменяет движок проверки орфографии по умолчанию на собственные словари вместе с функцией проверки грамматики. Вы догадаетесь, что он работает, по синему подчеркиванию мест пропуска запятых, неверных предлогов, повтора слов и прочих огрехов. Разумеется, он навряд ли сделает ваш текст столь же блистательным, как после профессионального редактора [Ред.: — Ну и в чем тогда счастье?!], но, по крайней мере, вы избежите наиболее частых ошибок. Каждый новый релиз *LanguageTool* предлагает больше грамматических правил и обогащает имеющиеся словари новыми терминами и фразами, и версия 3.7 — не исключение. Обязательно обновите его, если вы активно используете *Writer!*

Настройка движка браузера

Webenginepart

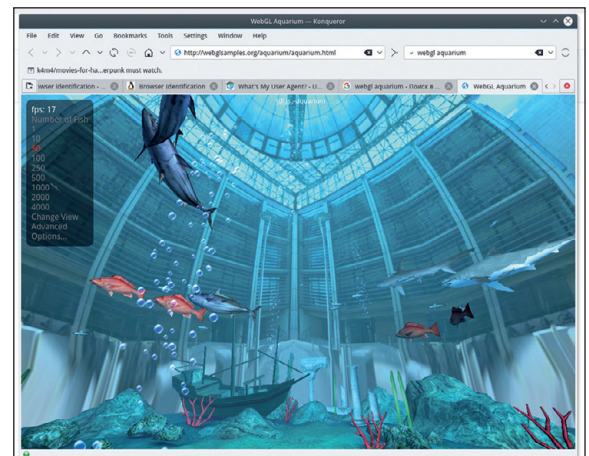
Версия: 17.04 Сайт: <http://bit.ly/2p9Aq0d>

Когда вы будете читать эти строки, уже выйдет релиз KDE Applications 17.04, вносящий новые функции в стандартный браузер KDE Plasma, известный как *Konqueror*.

Не секрет, что в последние годы *Konqueror* не привлекал особенного внимания ни со стороны разработчиков, ни со стороны пользователей, почему и плелся позади *Chrome* и *Firefox*. С технической точки зрения причиной является устаревший движок рендеринга HTML. Раньше у *Konqueror* имелось две опции: очень старый KHTML, ответивший от Apple в 2002 г. и ставший основой для *Safari*, и более новый и улучшенный Webkit, который всё равно плоховат для современной работы в Сети. *Webenginepart* предлагает поддержку QtWebengine для приложений KDE, и больше всех от этого выигрывает *Konqueror*. *Webenginepart* заменяет *KWebkitpart* и позволяет *Konqueror* применить современный движок рендеринга при сохранении

всех своих расширенных функций. В настоящий момент он соединяется с браузером с помощью QtWebengine 5.8, который внутри содержит слегка модифицированный движок *Chromium* 53.0.2785.148. Не самая последняя версия, но всё равно очень достойно. Вы можете выбрать новый движок в меню Settings > Configure Konqueror > General и немедленно насладиться колоссальной скоростью и совместимостью в вашем новом работающем на *Chromium* браузере *Konqueror*. Среди особо заметных изменений — плавное воспроизведение видео, возможность изящной обработки веб-страниц с изобилием JavaScript и намного лучшая производительность при работе с несколькими вкладками.

Среди заметных изменений — плавное воспроизведение видео.



Это солидное улучшение для *Konqueror*.

Webenginepart существует уже не первый год, но более ранние реализации в начале 2016 г. были недостаточно стабильны и слишком часто приводили к сбоям браузера, в противовес браузерам, использующим QtWebengine напрямую, как, скажем, браузер *Qupzilla*. Но за последние месяцы многие из досадных ошибок наконец-то исправили.

Повторим, *Webenginepart* — это только посредник, а вовсе не сам движок браузера, поэтому именно будущие обновления QtWebengine на самом деле улучшат вашу работу с *Konqueror*.

HotGames Развлекательные приложения

Игра-аркада

Man vs Olives

Версия: f8143a Сайт: github.com/pekupk/manvsolives

Не спрашивайте, зачем (и как) людям надо сражаться с оливками; это просто следует воспринимать как должное, как будто вы смотрите боевик по некой не очень логичной книге.

Вы управляете человеком в обстановке трехэтажных платформ, под которыми вода. Ваш персонаж может передвигаться в разные стороны, прыгать и даже заходить в воду, чтобы собирать падающие монеты. Здесь также есть драгоценности, которые помогают вам набирать очки. Собрав все драгоценности данного уровня, вы получаете ключ, который переводит вас на следующий уровень.

Но сверху падают не только монеты: злые зеленые оливки на первом уровне и жуткие черные маслины на более высоких градах сыплются вокруг вас. Как ни странно, они вас не трогают, когда вы их касаетесь либо перепрыгиваете через

них; вы даже можете столкнуться с оливками, чтобы они разлетелись по разным направлениям, как камни для керлинга. Но если какая-нибудь оливка шандарахнет по вам сверху — вы покойник.

Впрочем, можно умереть еще одним способом: перейти на самый низкий уровень и подождать, чтобы по вам пробежалась розовая свинья.

Если всё описанное выше показалось вам сущей нелепостью, значит, вы всё прочитали правильно. То, что приключается в *Man vs Olives*, явно не от мира сего. Но это также весело и до странности интересно, с уютной графикой и чувством

Не спрашивайте, зачем (и как) надо сражаться с оливками.



► Бегайте. Прыгайте и охотитесь за монетами, избегая зеленых и черных оливок. Да, именно так.

юмора во всех деталях. Игра разрабатывается в Финляндии и напоминает нам *Oilwar* из LXF204. Обе игры невероятно просты, но не примитивны, и очень захватывают. Так что если вам хочется провести полчаса за игрой, мы настоятельно рекомендуем *Man vs Olives*.

Игра отлично пойдет локально после того, как вы скачаете tar-архив с Github, или вы можете сыграть в онлайн-версию на <https://softcubicle.com/manvsolives>.

Игра-стрелялка

Tank Island

Версия: GIT Сайт: <http://bit.ly/2p01IGF>

Хотя через минуту-другую мы уже начнем стрелять по вражеским танкам, всё же для начала пара слов, чтобы ввести вас в курс дела. *Tank Island* — рабочий пример игры, разработанной на языке Ruby и пошагово описанной в книге *Developing Games With Ruby* [Разработка Игр на Ruby], причем они обе написаны Томасом Варанецкасом [Tomas Varanecas], горячим энтузиастом Ruby. Это стрелялка с видом сверху, созданная с помощью Gosu, библиотеки разработки 2D-игр для Ruby и C++, поэтому дело здесь не только в игре, но и в программировании. Если вам понравится *Tank Island*, вы можете попробовать сделать похожую игру самостоятельно. И вы не найдете документации к игре лучше, чем толстое руководство для начинающих на Ruby.

Однако вернемся к игре.

Вы управляете танком на стрелковом полигоне среди острова, который покрыт

травой и лесом и имеет песчаную береговую линию. Ландшафт передан в общих чертах и довольно грубо — каждая плитка мозаики являет собой всего-навсего квадрат определенного типа без скругления углов. Но графика более разнообразна благодаря тонким деталям разных объектов на поле — например, топливных баков, ангаров и диких кустарников. Суть игры заключается в том, чтобы провести ваш танк с помощью клавиш WASD и нацеливаться и стрелять по другим танкам с помощью мыши. В общей сложности на поле восемь игроков: это вы и семь ИИ-ботов. Сначала, пока вы не приобретете определенных навыков выживания, игра

Боты пытаются вас подстрелить, когда вы в досягаемости.



► Будьте осторожны и стреляйте на поражение до того, как это сделает вражеский танк.

проводит вас по бесконечному циклу постоянно множущихся случайно выбранных мест на острове, с врагами-ботами, которые пытаются вас подстрелить, когда вы окажетесь в пределах досягаемости. Через некоторое время вы, вероятно, выработаете стратегию более осторожного подхода и упреждающей стрельбы — это ключ к набору большего количества очков, чем боты. На поле также попадают полезные бонусы, которые могут восполнить ваше здоровье, улучшить скорость и увеличить дальность стрельбы. Выживание в танковой дуэли — это испытание своих сил и удовольствие.

Текстовый редактор

GNU Nano

Версия: 2.8 Сайт: <https://nano-editor.org>

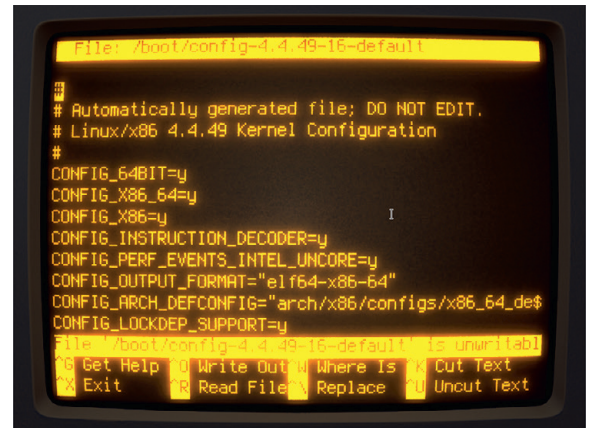
Мы полагаем, что постоянные пользователи Linux обязаны быть хорошо подкованы в использовании редакторов командной строки, независимо от того, какой это редактор.

Исторически основными редакторами консоли для систем на основе UNIX были *Vi* и *Emacs*, и оба до сих пор широко используются ветеранами Linux, но в наши дни часто вызывают негативные чувства из-за односложности *Vi* и необходимости учить сочетания клавиш *Emacs*; поэтому многие счастливы работать с *Nano*. *Nano* — куда более простой текстовый редактор с дружелюбным интерфейсом и чисто базовыми функциями (к примеру, по сравнению с *Vi* или применением *Emacs* в качестве целой системы *init!*). И это очень неплохо, если прибегать к редакторам командной строки лишь в особых случаях — скажем, для редактирования файла `etc/sudoers` или исправления файла настройки, когда ваша система почему-либо не загружается в графический режим.

В *Nano* не имеется «командного режима», и в этом он похож на классические текстовые редакторы DOS или приложения типа *mcedit*. Нижняя часть экрана отображает ссылки на популярные действия, всегда оставаясь у вас перед глазами. Минимум знаний, необходимых для работы с ним, таков: `Ctrl+O` для сохранения изменений и `Ctrl+X` для возвращения в консоль.

Есть масса горячих клавиш для навигации по строкам, символам и словам в тексте, по большей части они привязаны к `Alt` или `Ctrl+<что-то>`; но неплохо знать, что *Nano* может эмулировать сами `Alt` и `Ctrl`. `Alt` — это `Esc`, а `Ctrl` — двойной `Esc` (`Esc Esc`); в ряде случаев это дает больше гибкости.

Новый релиз *Nano* 2.8 отмечен переходом на библиотеку *Gnutilib*, что вовлекает



► Прimitивный в глазах седобродых, но очень мил для остальных!

кучу изменений. Пользователи заметят улучшение навигации по слишком длинным строкам, которые не умецаются на экране. Стрелки вверх и вниз вместе с клавишами `PgUp` и `PgDown` теперь перемещают вас не по логическим строкам, а по настоящим.

Клавиши `Home` и `End` также в первую очередь касаются видимых строк. При наличии строки очень большой длины первое нажатие на `Home` или `End` переводит в начало или конец видимой строки, а повторное нажатие — в начало/конец логической строки. Чтобы протестировать эту функцию, надо написать длинную команду или взять экран меньшего размера — отличной площадкой будет *Cool Retro Term!*

Масса клавиш для навигации по строкам, символам, словам.

Организатор файлов

Classifier

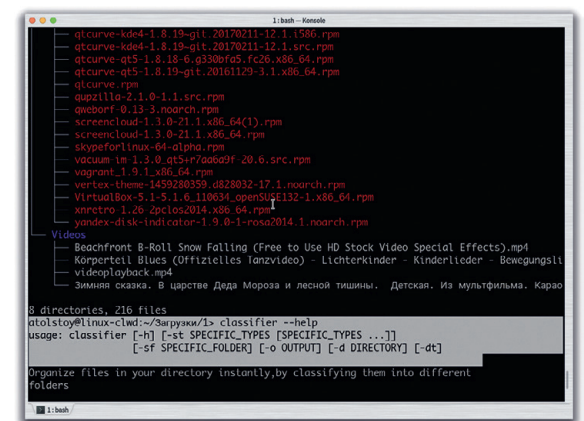
Версия: GIT Сайт: github.com/bhrgu123/classifier

Часто ли вы проверяете свою директорию `~/Downloads` на предмет наличия старых, забытых, повторяющихся или иных ненужных зачек? Многие пользователи Linux вообще не думают о том, чтобы разобраться с такими файлами, и просто копят всё, что скачали, пока их раздел `$HOME` не заполнится до отказа. Хорошо перфекционистам, у которых всё в идеальном порядке; но *Hotpicks* предназначены для остальных. Мы обнаружили скрипт Python, который классифицирует файлы по категориям в соответствии с типом файла. Пускай *Classifier* — простой и аскетичный инструмент объемом всего несколько килобит, но он делает жизненно важную работу. Перейдите в место, похожее на книжный шкаф со сломанными полками, и затем запустите инструмент без всяких аргументов: `$ classifier`.

Команда завершится практически сразу (поскольку перемещение файлов внутри одного раздела в Linux не требует времени),

и вы увидите, что теперь ваша директория содержит намного меньше файлов (если вообще содержит), зато много дополнительных поддиректорий. Мы использовали нашу тестовую директорию `~/Downloads` в качестве классического примера, и признаем, что *Classifier* применяет очень толковую и адекватную сортировку. Так, он определил RPM- и Deb-пакеты как отдельные категории, сгруппировал все архивы в одну поддиректорию и рассортировал изображения, документы, музыку и даже файлы EXE. Здесь нет функции `undo`, так что, возможно, вам стоит скопировать свою директорию и запустить на ней *Classifier* в качестве сухого прогона. В любом случае, данные не пропадут, но вам, возможно,

Classifier применяет толковую и адекватную сортировку.



► Идеальная уборка безо всяких усилий возможна!

понадобится дополнительное время на розыск своих файлов после того, как *Classifier* наведет там свои порядки. Установка инструмента исключительно проста, поскольку *Classifier* включен в онлайн-репозиторий модулей Python, и делается так:

```
$ sudo pip install classifier
```

Как и следует ожидать, можно прямо указать директории ввода и вывода и определить типы файлов вручную — все подробности вы найдете в `$ classifier --help`. LXF

На диске

Дистрибутивы, приложения, игры, книги и всякое-разное...

Лучшее из Интернета, упакованное в 8 ГБ качественного DVD.



Дистрибутивы

Одна из проблем, с которыми сталкиваются коммерческие программы — даты релизов. Если вы заготовили экстравагантную маркетинговую кампанию со стартом в назначенный день, лучше вывести свой продукт на рынок, даже если он еще не готов — вы всегда сможете исправить ошибки потом (Service Pack переводится как «Маркетинг не дал нам это сделать вовремя и нормально»). Программы с открытым кодом часто горды своим выходом «по мере готовности», хотя нередко это выливается в досадные задержки. Тщательный процесс тестирования Debian означает, что мы ждем нового релиза невероятно долго, и ко времени выхода он уже устарел; зато можно поспорить, что ваша работа на нем будет стабильна, как и делают многие сисадмины.

Правда, Ubuntu дважды в год показывает, что держаться расписания вполне реально (хотя и у них бывают задержки сроков). Всё, что тут нужно — дисциплина и автократичное управление. Разработка комитетом имеет свои преимущества, но иногда хорошо, если кто-то скажет: «Значит, так. Никаких новых фишек в этом релизе». Ничто не вредит расписанию релизов больше, чем «ну вот только еще одно» — оставьте это детективу Коломбо.

Neil

» Важно ВНИМАНИЕ!

Порченные диски

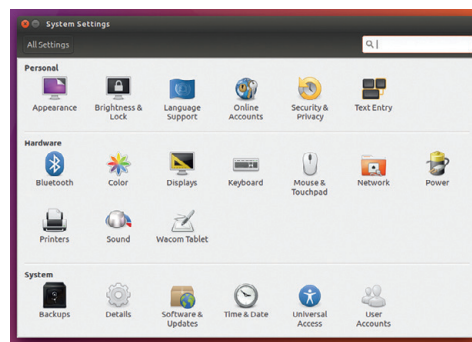
В маловероятном случае какого-то дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу disks@linuxformat.ru или телефону +7 (812) 309-0686.

Самый свежий релиз

Ubuntu 17.04

Дважды в год огромное число пользователей Linux нетерпеливо ждут нового релиза Ubuntu. С регулярностью часового механизма каждый апрель и октябрь устройства для прожига DVD во всем мире разогреваются в предвкушении. Несмотря на периодические повороты в ложном направлении и даже то, что они отпугнули немало постоянных пользователей употреблением Unity, Ubuntu умудряется сохранять популярность среди обычных пользователей Linux. На то есть серьезная причина: он просто работает. Для тех, кто привык исключительно к Windows, Ubuntu совсем иной, и всё же он прост в работе (мы не говорим, что для других дистрибутивов это не так); а когда людям что-то нравится, вряд ли они откажутся от этого без оснований.

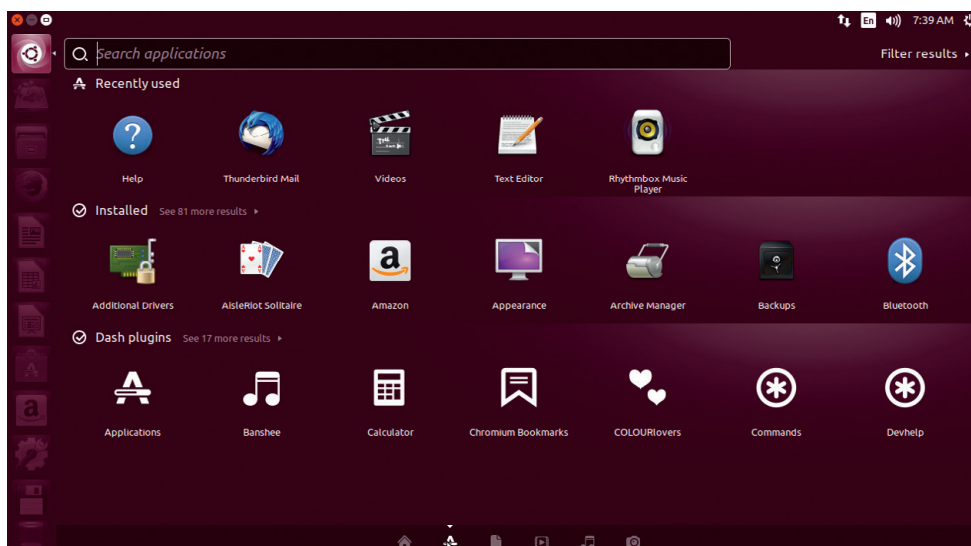
На нашем DVD помещена базовая (в смысле отсутствия излишеств) Ubuntu 17.04 в 32- и 64-битной версии. Для этого нам пришлось до последнего отодвигать срок производства, так что у нас нет ремикса рабочих столов. 32-битная версия будет работать на любом стандартном ПК (с поддержкой PAE) за последние 20+ лет. Это делает его самым переносимым, но зато лишает преимуществ современных процессоров. Для них нужна 64-битная версия, и этот вариант вам и стоит использовать, если ваша машина его поддерживает. Если вы не уверены в том, что у вашего компьютера внутри, загрузитесь с DVD. Если меню загрузки для 64-битной версии показывает предупреждение, что оно не будет работать на 32-битном оборудовании, значит, у вас 32-битное оборудование. Предупреждение



» Вы можете изменять настройки при загрузке с DVD, но они будут утрачены при перезагрузке. Установите его, и вы сможете настроить все по своему желанию.

показывается, только если оно касается вашей системы. Если вы его не видите, значит, можете спокойно загружать 64-битную версию. DVD должен загрузить рабочий стол Unity, откуда можно делать всё что угодно. Вы можете работать с Ubuntu в этом режиме live, однако при перезагрузке все внесенные вами изменения будут утрачены; или установите его на жесткий диск. Помните, что при работе с DVD система будет тормозить — это из-за времени, требуемого на считывание данных с DVD. Скопировав DVD на флешку USB, как подробно описано в FAQ на DVD, вы сильно ускоритесь.

Информация для входа в систему: имя пользователя — **ubuntu**, пароль — пустой.



LIVE-ДИСК: ЗАГРУЗИ > ОПРОБУЙ > УСТАНОВИ!

Ubuntu

17.04

Новинка-2017!

- » Ядро 4.10
- » Рабочий стол Unity 8
- » Тысячи приложений!

32- и 64-битный



Новичок в Linux? Начните отсюда!

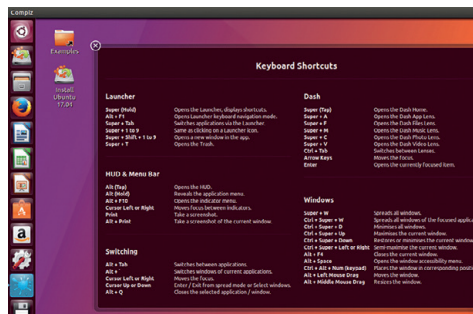
- » Что такое Linux? Как его установить?
- » Есть ли в нем эквивалент *MS Office*?
- » Зачем нужна командная строка?
- » Как устанавливать программы?

Ответы приводятся в [Index.html](#) на диске.

Для более старого оборудования

Ubuntu 17.04

Ubuntu — один из тающего числа дистрибутивов, которые продолжают выпускать 32-битные диски и программы установки, и эта версия пригодится вам для более старого оборудования. Дистрибутив должен загрузиться прямо в рабочий стол. Мы предлагаем информацию для входа в систему из среды live, но если вам предложат ввести логин при загрузке, значит, что-то пошло не так. Обычной причиной является отсутствие драйверов для вашей видеокарты, но взгляните также в FAQ на DVD. Информация для входа в систему: имя пользователя — **ubuntu**, пароль — пустой.



Легковесная редакция

Linux Lite 3.4

Здесь у нас быстрый и легковесный дистрибутив рабочего стола на Ubuntu, но с *Xfce* вместо Unity. Это не единственное отличие: есть ряд других улучшений и изменений. Linux Lite подойдет новичкам, но хорош и для более опытных членов сообщества Linux. К тому же для дистрибутива, сфокусированного на производительности, он очень мило выглядит. Выбор рабочего стола *Xfce* позволяет добиться неплохого компромисса между скоростью и функциональностью, предоставляя все те функции, которыми пользуется

большинство, но при этом сохраняя быстрый и отзывчивый рабочий стол.

То, что Linux Lite основан на Ubuntu, дает ему и другим подобным дистрибутивам несколько преимуществ. У основного дистрибутива солидная, хорошо протестированная база, а кроме того, имеется широкий выбор пакетов из репозитория Ubuntu и множество PPA, которые вы можете добавить, если учтете в себе дух авантюризма. Информация для входа в систему: имя пользователя — **linux**, пароль — пустой.



И еще!

Системные инструменты

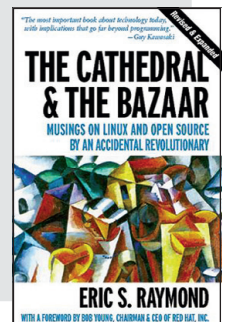
Главное

- 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 самого последнего стабильного релиза ядра.
- Memtest86+** 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 Словях]** Введение в ядро, написанное великим магистром ядра Грегом 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!

LXF223 Май 2017



- » Ставим на Zero Raspberry Pi Zero W уже с Wi-Fi и Bluetooth
- » Дизайн для 3D-принтера Создание моделей
- » В стиле Blu-ray Заставим работать в Linux
- » Стеганография И чтоб никто не догадался

LXFDVD: feren OS, IPFire, openSUSE Tumbleweed, RancherOS, Rescatux, Ubuntu Studio, XenialDog, 12 книг о Linux, горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_223/
PDF-версия: shop.linuxformat.ru/elxf_223/

LXF222 Апрель 2017



- » 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, горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_222/
PDF-версия: shop.linuxformat.ru/elxf_222/

LXF221 Март 2017



- » Прощай, Google! Мы найдем получше
- » Ваш добрый демон Выбор ОС BSD
- » Сервер на халюву CentOS, клон Red Hat
- » Linux + Windows Сожительство возможно

LXFDVD: ExTiX, Porteus, Rebecca Black Linux, Siduction, Ubuntu, Zorin, 12 книг о Linux, горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_221/
PDF-версия: shop.linuxformat.ru/elxf_221/

LXF219/220 Февраль 2017



- » 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, горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_219-220/
PDF-версия: shop.linuxformat.ru/elxf_219-220/

LXF218 Январь 2017



- » 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, PelicanHPC, Ubuntu, 12 книг о Linux, горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_218/
PDF-версия: shop.linuxformat.ru/elxf_218/

LXF217 Декабрь 2016



- » Сплошная малина Всё про Raspberry Pi
- » Болтунам Без находок для шпиона
- » Блогодеяния Свежая WordPress
- » Интернет вещей Строим домашних, построим и дом

LXFDVD: Android-X86, Elementary OS, Korora, Mint KDE, Maui, 12 книг о Linux, горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_217/
PDF-версия: shop.linuxformat.ru/elxf_217/

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
 DVD
 ФОРМАТ

Страница 1

ДИСТРИБУТИВЫ

Ubuntu 17.04 (64-битный)
Ubuntu 17.04 (32-битный)

Популярная ОС Linux для настольных компьютеров, серверно доступная с поддержкой сообщества и профессионалов. Согласно Манифесту Ubuntu, ПО должно быть доступно, программы интуитивно должны предоставляться людям на их родном языке и неварра на ограничения их физических возможностей, и пользователи должны иметь право настраивать и изменять свое ПО любым способом на свое усмотрение.

Linux Lite (32-битный)

Дистрибутив на базе Ubuntu, быстрый и легкий весный за счет рабочего стола Xfce вместо Unity.

ПОМОЩЬ

Руководство новичка
 Руководства
 Ответы
 Чаво (FAQ)

НОТИСЫ

Classifer (GIT) Органайзер файлов
github.com/brncln123/Classifier

GNU nano 2.8 Текстовый редактор
nano-editor.org

Genzli (GIT) Кодировщик изображений
github.com/drozdov/genzli

Languiaceto 3.7 Программа проверки орфографии
languiaceto.org

Man vs Olives 04.42a Игра-аркада
bit.ly/2n1LwWdK

Male 11.8 Средства рабочего стола
male-desktop.org

Mele-01 0.9.5 Приложение прогноза погоды
bit.ly/2nrd0dM

MIRPaint 3.50 Программа рисования
github.com/Amadeal

MTSS-3G 2017.23 Драйвер файловой системы
bit.ly/2n3r0dD

Tank Island (GIT) Игра-стрелка
bit.ly/2n01tGE

WebenGiferant 17.04 Настройка движка браузера
bit.ly/2n9A0dD

УЧЕБНИКИ

Fraclais
 Garbiviz
 Ghesvi

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

LIVE-ДИСК: ЗАГРУЗИ > ОПРОБУЙ > УСТАНОВИ!

Ubuntu

17.04

Новинка-2017!

- » Ядро 4.10
- » Рабочий стол Unity 8
- » Тысячи приложений!

32- и 64-БИТНЫЙ



Внимательно прочтите это перед тем, как использовать LXF DVD!

Страница 2

ГЛАВНОЕ

Checkinstall
Coreutils
Hardinfo
Kernel
Memtest86+
Plop
SBM
WvDial

ДОКУМЕНТАЦИЯ: 12 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

Advanced Bash Scripting Guide Подробное руководство по программированию на Bash
Bash Guide for Beginners Руководство по Bash для начинающих
Bourne Shell Scripting Начальное руководство по программированию на Bash

The Cathedral and the Bazaar Классический текст Эрика Реймонда (Eric S Raymond) «Собор и базар» о преимуществе открытой разработки

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 специальную терминологию

Linux Kernel in a Nutshell Описание ядра Linux, созданное одним из его выдающихся разработчиков — Грегом Кроа-Хартманом (Greg Kroah-Hartman)

Rule Знаменитый спрочачник
System Administrators Guide Руководство по базовому администрированию Linux

GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

ДИСТРИБУТИВЫ

С РАБОЧИМ СТОЛОМ KDE

Manjaro 17.0 (64-битный)
Быстрая и удобная настольная ОС на базе Arch Linux. Ключевые свойства — интуитивно понятный процесс установки, автоматическое обнаружение оборудования, стабильная возобновляемая версия, возможность установки нескольких ядер, специальные скрипты Bash для управления графическими драйверами и исключительная настраиваемость.

Netrunner 17.01.2 (64-битный)
Дистрибутив уникального внешнего вида, на базе Debian, с настраиваемым рабочим столом KDE и дополнительными приложениями, мультимедиа-кодеками, Flash и Java-плагинами. Эти изменения призваны повысить удобство среды рабочего стола при сохранении свободы настройки.

Все дистрибутивы представлены ISO-образами, которые можно записать на отдельный носитель, и загрузить в live-режиме прямо с LXF DVD. У всех присутствует возможность установка на жесткий диск.

Пожалуйста, перед использованием Асдного Диска ознакомьтесь с инструкцией, опубликованной в журнале на стр. 109!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДВОЕКЛЕТНЫЕ ДИСКИ В многократном случае обнаружения дефектов на диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аппаратный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных в вашем жестком диске. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, предоставленных нашей программой или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, скачайте программу для загрузки с ее веб-сайта: www.linuxformat.ru

Тираж издательства ООО «Марком», 186852, Россия, Ленинградская область, Всеволожский р-н, дер. Юрки, Школьная ул., 7-а. Лицензия ИПТР ВАО 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 Combus! 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



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № Ф077-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж печатной версии 1200 экз., распространение электронной версии 30000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Елена Ессяк, Даниил Кривошеин, Светлана Кривошеина, Валентин Развозжаев, Валерий Смирнов, Елена Толстякова, Ирина Шулакова

Редактор диска

Александр Баракин

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Андрей Смирнов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатель

ООО «Линукс Формат»

Отпечатано в типографии ООО «ЛД-ПРИНТ»

196644, Санкт-Петербург, Колпинский р-н, пос. Саперный, территория предприятия «Балтика», д. 6/н, лит. Ф
Тел. (812) 462-8383, e-mail: office@ldprint.ru
Заказ 15986

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор neil.mohr@futurenet.com

Научный редактор Джонни Бидвелл Jonni.Bidwell@futurenet.com

jonni.bidwell@futurenet.com

Выпускающий редактор Крис Торнетт Chris.Thornett@futurenet.com

chris.thornett@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоза

efrain.hernandez-mendoza@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Джонни Бидвелл [Jonni Bidwell], Нейл Ботвик [Neil Bothwick], Джейсон Кэннон [Jason Cannon], Нейт Дрейк [Nate Drake], Кент Ельчук [Kent Elchuk], Ти Джей Хейфер [T J Hafer], Джон Лэйн [John Lane], Джо Осборн [Joe Osborne], Ник Пирс [Nick Peers], Лес Паундер [Les Pounder], Афан Рехман [Afan Rehman], Томас Рамболд [Thomas Rumbold], Дэнниел Сэмюэлс [Daniel Samuels], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Александр Толстой [Alexander Tolstoy], Михалис Цукалос [Mihalis Tsoukalos], Евгений Балдин, Андрей Гондаренков, Алексей Федорчук, Максим Черепанов

Иллюстрации Шейн Коллиндж [Shane Collinge]

Иллюстрация с обложки www.magictorch.com

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Тел. +44 (0) 1604 251045, email: linuxformat@myfavouritemagazines.co.uk

РОССИЯ: Санкт-Петербург, пр. Медиков, 5, корп. 7

Тел. +7 (812) 309-0686, e-mail: info@linuxformat.ru

По вопросам сотрудничества, партнерства, оптовых закупок:

partner@linuxcenter.ru

Авторские права: статьи, переведенные из английского издания Linux Format, являются собственностью или лицензированы Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на 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 повреждение или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

GNU/Linux — заменяется на "Linux" в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журнале, издаваемом Future plc group company, обращайтесь на сайт www.futureplc.com.



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

16+

В июльском номере

Переиздание 2017 г.:

Изучаем взлом

Облачайтесь в костюм «белого» хакера: тестируем наши сети на вторжение, чтобы убедиться в их непробиваемости.

Linux на MacBook

Гибрид старого MacBook и Linux предоставит вам лучшую ОС на лучшем оборудовании.

Ваш личный Интернет

Создайте себе потайные и безопасные сайты, чтобы делиться секретами при надежном их сохранении.

Брякаем ключами!

Рассматриваем, как устройство Yubikey обеспечивает нашу безопасность, и скрытно беседуем, и не только, онлайн с помощью Keybase.

Содержание будущих выпусков может меняться — вдруг мы ввиду паранойи побоймся вылезти из-под одеяла...

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

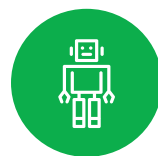
- Праздничные мастер-классы
- Городской лагерь на время каникул
- Курсы повышения квалификации для педагогов



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

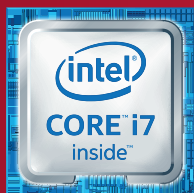
ОСТРЫЙ КАК ЛЕЗВИЕ

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 1080
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.