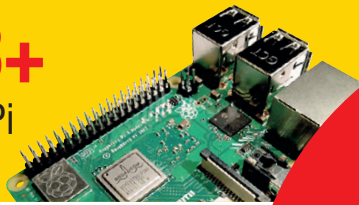


НОВЫЙ!

Raspberry Pi 3 B+

Все подробности о всемогущем Pi и учебный проект для новичка!



**ДАРИМ
8-ГБ DVD!**
ЛУЧШЕЕ ИЗ ARCH

- ARCH
- MANJARO
- ARCH 32 & ЕЩЕ

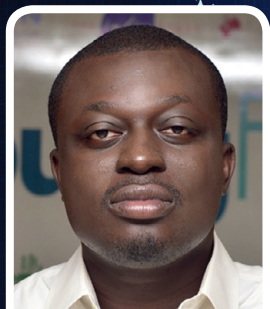
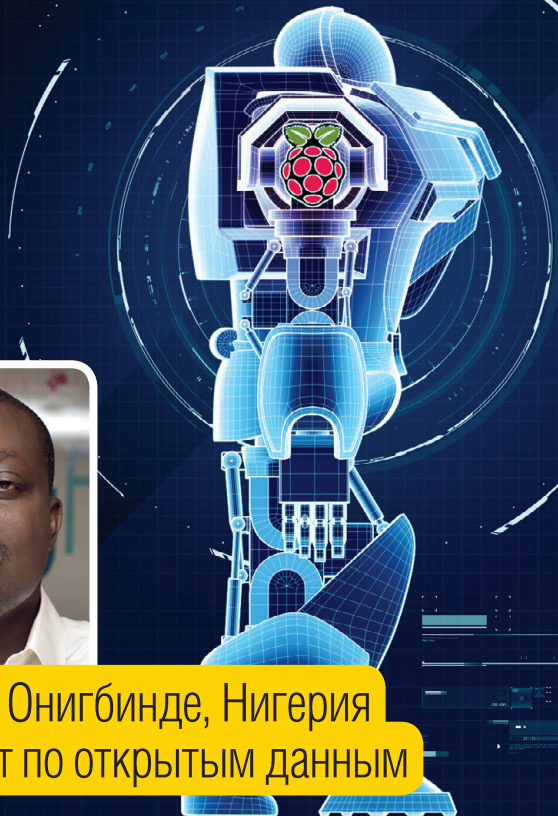


LINUX FORMAT

Главное в мире Linux

RASPBERRY Pi: РОБОТЫ!

Учимся проектировать, собирать и программировать замечательные машины на Pi под управлением ИИ



Олусеун Онигбинде, Нигерия
Активист по открытым данным

ПРОЕКТЫ!

ПРИСТУПИМ
К МАШИННОМУ ОБУЧЕНИЮ

СОБЕРЕМ ФАЙЛ-СЕРВЕР,
БЛОКИРУЮЩИЙ РЕКЛАМУ

СОЗДАДИМ СВОЮ
ТЕМУ WORDPRESS

СЕТЕВОЕ ХРАНИЛИЩЕ

Лучшие открытые решения для совместного пользования файлами

УДАЛЕННЫЕ ВЫЧИСЛЕНИЯ

Распределите нагрузку, чтобы все ваши ПК были при деле

МАЙ 2018 LXF236



4 607130 825167 1 8005



iTeleRadio

ИНТЕРАКТИВНОЕ РАДИО&ТЕЛЕВИДЕНИЕ

iteleradio.ru



**LINUX
FORMAT**



» КТО МЫ

На сей раз мы спросили наших экспертов: мы создаем роботов на Raspberry Pi; а какую задачу PiBot поможет вам выполнять? Или он лишит вас работы?



Джонни Бидвелл

Хорошо бы робот написал несколько вопросов для недавно возрожденных Ответов. А другой робот — еще и ответил на эти вопросы, поскольку мне хватает и своих проблем с Linux. И знаете что? Вообще-то я скучаю по моему роботу DiddyBorg. Ну почему мне больше не выдают аппаратуру для развлечения?



Шон Д. Конвэй

PiBot, способный покормить моего пса. В теории это достаточно просто. Увы, чтобы оно сработало, PiBot должен быть моим двойником. У этого животного — импринтинг меня как единственного подателя пищи, и он отказывается признавать кого-либо другого. Вот вам собачья верность, чтоб ее...



Мэтью Хэнсон

Я с завистью наблюдал за одним из этих роботов-пылесосов. Всё, что может помочь очистить мой дом, набитый домашними животными, и высвободить мне время под игры с моей новой Raspberry Pi и игры на Linux, для меня правильная инвестиция!



Лес Паундер

Слышал, что американская забегаловка с гамбургерами установила первый робот — подавальщик гамбургеров на полный рабочий день. Skynet умна. Она хочет откормить нас вкусными гамбургерами, чтобы нас было легче поймать из-за недостатка физических упражнений. Умный ИИ от Skynet!



Шашанк Шарма

Сомневаюсь, что PiBot сможет волюнтерить не хуже, чем я, шлифовавший это искусство годами, так что мне нечего бояться. Правда, автоматизировали составление контрактов и соглашений, что угрожает моей основной работе. Но я, конечно, не предвижу никаких ботов, пишущих для этого замечательного журнала.



Валентин Синецын

PiBot гораздо лучше меня умеет вычислять число π. Не то чтобы мне приходилось делать это ежедневно, но когда мне понадобится стотомиллиардный знак после запятой — я уж знаю, куда нужно обратиться. Ну и раз никто еще не сказал эту избитую шутку: варить кофе.



**Подпишись
& сэкономь!**

Как это сделать?
См. с. 3

«Вкалывают роботы...»



«Счастлив человек...» Вторая часть этого «прогноза» из фильма «Приключения Электроника», вышедшего на экраны в 1980-м, пока не сбылась. А вот первая вполне осуществилась. Появление доступных по цене микроконтроллеров и шаговых электродвигателей привело к повсеместному использованию промышленных роботов, станков с ЧПУ и 3D-принтеров, позволив перейти от поточного производства одинаковых изделий к гибкому и индивидуальному. Экономисты уже назвали это очередной

промышленной революцией. Появление микропроцессоров, надежной цифровой радиосвязи, микромеханических датчиков, легких литиевых батарей и бесколлекторных электродвигателей позволило создать множество мобильных роботов, работающих в воздухе, на поверхности и под водой. Это уже привело к революции в военном деле, а многие гражданские применения упрутся скорее в юридические, а не технологические барьеры. Пример — доставка посылок по воздуху. Сегодня ко всему перечисленному добавились недорогие системы технического зрения и нейронные сети. Роботы стали заменять человека там, где требуется распознавание образов и принятие более-менее стандартных решений. Пример — управление автомобилем на дорогах общего пользования. Но самое главное, на мой взгляд, даже не это. Создание роботов стало популярным видом технического творчества. Этому даже в приличных школах учат. Надеюсь, что для многих современных детей робототехника станет любимым делом, а значит, дальнейшие успехи гарантированы.

Кирилл Степанов, главный редактор
info@linuxformat.ru

» КАК С НАМИ СВЯЗАТЬСЯ

Письма для публикации: letters@linuxformat.ru

Подписка и предыдущие номера: subscribe@linuxformat.ru

Техническая поддержка: answers@linuxformat.ru

Общие вопросы: info@linuxformat.ru

Проблемы с дисками: disks@linuxformat.ru

Вопросы распространения: sales@linuxformat.ru

Сайт: www.linuxformat.ru, группа «ВКонтакте»: vk.com/linuxform

➤ **Адрес редакции:** Россия, Санкт-Петербург, пр. Медиков, 5, корп. 7

➤ **Телефон редакции:** (812) 309-0686. Дополнительная информация на с. 112

Знаете всё о Linux?

ДА

НЕТ

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

Зарегистрируйтесь
на сайте
shop.linuxformat.ru



СОМНЕНИЯ

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



Версия для iPad
и iPhone доступна
в App Store

Выберите вид подписки

PDF-версия
на 6 месяцев
1485 ₺

PDF-версия
на 12 месяцев
2760 ₺

Печатная версия
на 6 месяцев
2430 ₺

Печатная версия
на 12 месяцев
4500 ₺

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 с 2005 года
- Порадуются новинкам открытого ПО на DVD-приложении к Linux Format в каждом номере

Способы доставки

- Курьером по Москве и Санкт-Петербургу
- Курьером и в пункты выдачи iml.ru по всей России
- Почтой по всей России заказной или простой бандеролью
- Самовывоз из офиса «ГНУ/Линуксцентра» в Санкт-Петербурге

Способы оплаты

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



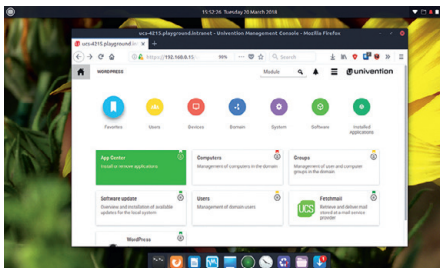
«ГНУ/Линуксцентр»
Санкт-Петербург,
пр. Медиков, 5, корп. 7
(метро «Петроградская»)
(812) 309-0686
www.linuxformat.ru

Содержание

ОБЗОРЫ

Univention Corporate Server 4.3 14

Контейнеры *Docker* и опрятное управление через Web превращают этот дистрибутив в лакомый кусочек для предприятий малого и среднего бизнеса.



Endless OS 3.3.11 15

Endless = бесконечный... а бесконечное — что? Наслаждение от дистрибутива или раздражение из-за продолжительной загрузки? Как-никак, 15 ГБ и Википедия впридачу...

AMD Ryzen 5 2400G 16

Наконец-то процессоры от AMD обзавелись встроенной графикой — и первый блин комом не вышел. Вы получите игры 1080р на 60 кадров в секунду, с восьмипоточной обработкой.



Seagate IronWolf 12TB 18

Жесткие диски не ржавеют! Количество дензнаков за петабайт отродясь не падало ниже! [Ред.: — Ну, это явная натяжка]. Мы сравнили новый экземпляр от Seagate с твердотельным вариантом: кто кого соберет?

Railway Empire 19

Здесь вам придется не просто играть в железную дорогу, а предварительно строить ее; в результате вы волей-неволей становитесь первоклассным управляющим.

RASPBERRY Pi: РОБОТЫ!

Мы занялись проектами роботов на Raspberry Pi, чтобы разобраться с управлением, подключением кода и сборкой самого крутого устройства, на с. 28



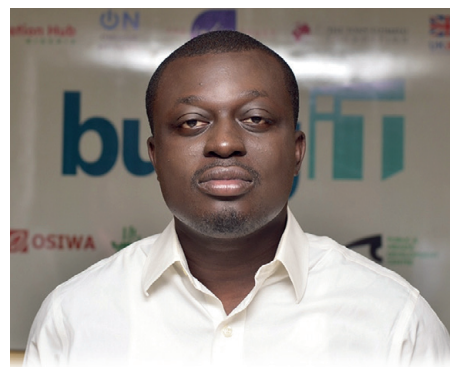
СПРАВНЕНИЕ



Дистрибутивы NAS 22

Если вы прикидываете, как обеспечить доступ ко всем своим данным через удобное центральное хранилище, то вам повезло! Мы предлагаем решения, достойные рассмотрения.

ИНТЕРВЬЮ

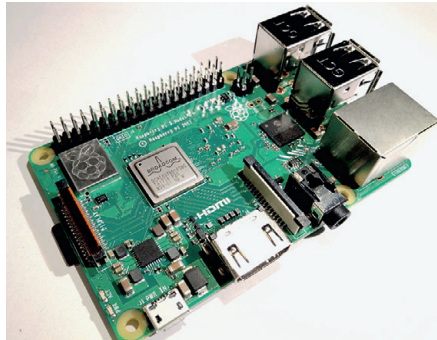
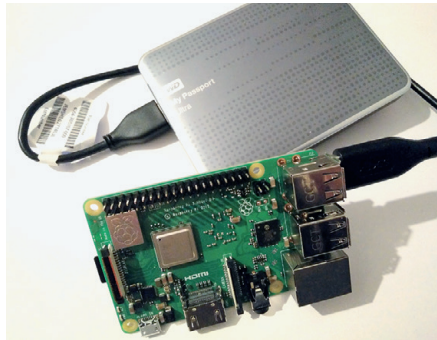


Олусеун Онигбинде держит правительство под контролем через открытость данных 38

ПОЛЬЗОВАТЕЛЯМ RASPBERRY PI

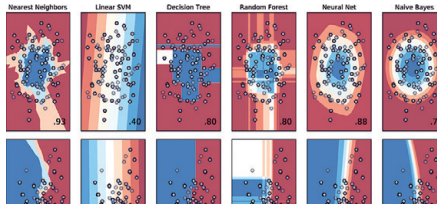


- Новости Pi** 83
Что предлагает гонщикам новый сезон Formula Pi, камера Pi ныряет под воду, а робокоту на Pi 3 этого совсем не хочется.
- Raspberry Pi 3 B+** 84
Радость в стане Pi! Pi Foundation раскурила свой Pi 3, увеличив тактовую частоту, улучшив работу с сетью и вставив новый SoC.
- Настраиваем сервер** 86
Превратите свой Pi в мощный сетевой файловый сервер и бдительный блокировщик рекламы. Идеально и для старых, и для новых Pi.
- Контроллер тепла** 88
Если вы проживаете в тундре, то вам очень пригодится отопительный блок под управлением Pi. Да и не обязательно в тундре.
- Собираем Raspbian** 92
Это чтобы решение получилось самостоятельным и максимально эффективным с точки зрения производительности.



АКАДЕМИЯ КОДИНГА

- Змеиное обучение** 74
Обработка данных, используя методы машинного обучения и Python 3.
- Шаблоны WordPress** 78
Всё, что вам надо знать для преобразования темы практически любого сайта в шаблон WordPress.



ПОСТОЯННЫЕ РУБРИКИ

- Новости** 6
Samsung выдал сверхбыстрые накопители для геймеров, а Intel — такой же чип и SSD, семья Ubuntu обновилась, хромбуки сроднились с Linux, развернут кластер на ARM, а AMD экономит энергию.
- Вести мобильных ОС** 20
Носимые устройства занялись спортом, Samsung открывает учебный центр в Перми и создает новые смартфоны, а Huawei готовит альтернативы Android и Windows.
- Сравнение** 22
Не можете решить, какой дистрибутив самый подходящий для управления вашим сетевым хранилищем? Наш ответ: EasyNAS, FreeNAS, NAS4Free, OpenMediaVault, Rockstor.
- Интервью LXF** 38, 42
Олусеун Онигбинде не дает спуску правительству, а Андрей Черепанов учит KDE говорить по-русски.
- Рубрика сисадмина** 50
Ресурсы бывают ограниченными, но как удержаться в рамках ограничений? Д-р **Валентин Синицын** знает способы умерить аппетиты процессов Linux благодаря cgroups.
- Ответы** 98
Чиним читалку RSS, подбираем менеджер паролей, употребляем диски ремиксов **LXF!**
- HotPicks** 100
Отведайте горяченького! Лучшие в мире новинки свободного ПО: *Bleachbit, Daemon-vs-demon, DDrescue, Falkon, Fstransform, Gnome Layout Manager, Polo, Stupax, Terminal, VLC, WINE.*
- Диск Linux Format** 106
- Предыдущие номера** 108
- Через месяц** 112

LINUX FORMAT 8-ГЕ

MANJARO

Дистрибутив Arch для Raspberry Pi

- Самообновляемый
- Мощный и дружелюбный
- Классический легковесный рабочий стол Xfce

ЗАДАРОМ НА DVD
Manjaro Linux 17.1 Arch Labs 2018 Arch 32, Arch Pi с. 106

НОВИНКА!

Arch 32 2018.02 Сборка для старых ПК 32-битный

ArchLabs 2018.03 Новый пацан во дворе 64-битный

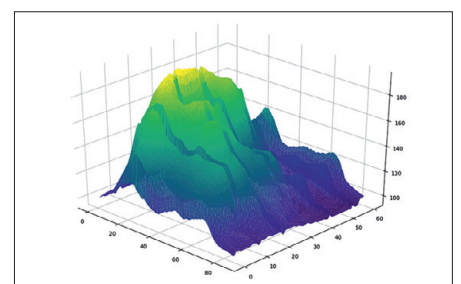
Arch Pi Для Pi 2/3 Оптимизирован под ARM 32-битный

УЧЕБНИКИ

- ТЕРМИНАЛ: Watson** 56
Отследить время, потраченное на выполнение проекта — Геркулесов труд. Но с *Watson* это элементарно.
- СЕТИ: Компиляция издали** 58
Распределенные вычисления трудоустраивают все ваши компьютеры преклонного возраста и разгружат молодых бойцов.
- OPENOFFICE: Офис на Nextcloud** 62
Отчего бы не завести облако на дому? Оно годится для хостинга корпоративных сайтов, размещения сервера электронной почты и, наконец, офисной работы.
- ОБЛАКА: Syncthing + бэкап** 66
Создаем архитектуру надежного зашифрованного облачного хранилища. Нужен только сетевой компьютер или вообще смартфон!
- NODE-RED: MicroPython** 70
Наша беспроводная Вещь излучает данные через LoRa, а мы эти данные собираем и превращаем в элегантные графики, понятные всем.

В ПОДРОБНОСТЯХ...

- Как обучаются машины** 46
Перед человечеством встает проблема умных машин. Разбираемся, как открытый искусственный интеллект завоевывает мир.



Новости

В ЭТОМ НОМЕРЕ: Безумная скорость >> 10 триллионов >> Личное >> Очередной накопитель >> Апрельский Ubuntu >> Linux — на хромбуки >> Супер-ARM >> Холоднее...

НАКОПИТЕЛИ

Геймеры, внимание!

Samsung представляет «безумно быстрые» SSD 970 Pro и 970 Evo.

Samsung анонсировала «молниеносно быстрое» дополнение своей линейки твердотельных накопителей. SSD-приводы Samsung 970 Pro и 970 Evo выполнены в форм-факторе M.2 2280 с интерфейсом PCIe Generation 3×4 lane, оснащены 64-слойной флэш-памятью 3D NAND и новым контроллером от Phoenix. Заявленные характеристики ошеломительны: 970 Pro обеспечивает 3500 МБ/с при последовательном чтении и 2700 МБ/с при записи; у Evo — 3500 МБ/с и 2500 МБ/с. Новый контроллер и V-NAND дали 30% ускорения против предыдущих моделей Samsung, и без того побивающих по этому параметру большинство конкурентов. Процесс записи ускорен за счет буфера на 78 ГБ (т. н. Samsung Intelligent

TurboWrite). Технология Dynamic Thermal Guard осуществляет автоматический мониторинг, поддерживая оптимальную температуру и защищая устройство от перегрева. «970-я серия во всех аспектах задает рынку SSD NVMe новую планку своей потрясающей производительностью, надежностью и лучшими в данном классе функциями», говорит Ун-Су Ким [Un-Soo Kim], старший вице-президент по бренд-маркетингу подразделения Memory Business в Samsung Electronics. Гарантия Samsung накопителям — 1200 ТБ записи (TBW) или 5 лет, в зависимости от того, что наступит раньше (для внутренних дисков это, вероятно, срок службы хост-устройства). Цены — \$329 для 512 ГБ Pro и \$629 для 1 ТБ. Моделей Evo больше: 250 ГБ за \$119, 500 ГБ за \$299, 1 ТБ за \$449 и 2 ТБ за \$849.



Даже внешне новые твердотельные накопители Samsung 970 Evo и Pro выглядят кусками кремния.

ПРЕДСТАВЛЯЕТ INTEL

«Это Матрица, брат!»

Чип от Intel выполняет 10 триллионов вычислений в секунду.

В силу своего уникального дизайна, решение Intel Stratix 10 на устройствах FPGA способно молниеносно выполнять сложные вычисления — порой в десятки и в сотни раз быстрее, чем микросхемы в потребительских продуктах. Новейшая версия, выпущенная в феврале этого года, обладает вычислительной мощностью 10 терафлоп или 10 трлн операций с плавающей запятой в секунду. Stratix 10 — самый быстродействующий в мире чип в своем классе.

FPGA-устройства, или программируемые пользователями вентиляционные матрицы — особый класс вычислительных чипов; они обретают особую важность с распространением приложений типа распознавание речи, технологии искусственного интеллекта, беспроводные

сети нового поколения, современные поисковые машины и системы высокопроизводительных вычислений.

В отличие от обычных процессоров (CPU), применяемых в современных ПК и ноутбуках, FPGA-устройства могут быть адаптированы — или удаленно, «на лету» перепрограммированы — для выполнения сугубо специализированных вычислительных задач.

FPGA Stratix 10 содержит около 30 млрд транзисторов; это в три с лишним раза больше, чем в процессорах, на которых работают самые быстродействующие ноутбуки и персональные компьютеры. Благодаря этому FPGA-устройства Intel Stratix 10 позволяют обрабатывать объемы данных, эквивалентные содержимому 420 дисков Blu-ray, всего за одну секунду.



Ребекка Невин [Rebecca Nevin], евангелист из университетской программы Intel FPGA, держит в руках матрицу Intel Stratix 10.

ПЕРСОНАЛИИ

Рыба ищет где глубже, а человек...

Джим Келлер, бывший архитектор CPU в AMD, переходит из Tesla в Intel.

Джим Келлер [Jim Keller], разработчик чипов AMD в наиболее успешные, вероятно, в истории этой компании времена, а затем ушедший в Tesla, теперь переходит в Intel. Уже последовало заявление от Intel, что Келлер получил должность старшего вице-президента и возглавит в компании подразделение по проектированию полупроводниковых схем, включая разработку и интеграцию систем-на-кристалле (SoC). Заявление Intel намекает на реорганизацию работы компании после успешного выпуска AMD Ryzen, последней линейки CPU на базе микроархитектуры Zen для настольных ПК и ноутбуков, которую Келлер разрабатывал до своего ухода в Tesla в 2015 г. «Мы приступили к инициативам, направленным на коренное изменение способов создания процессоров, когда мы входим в мир гетерогенных процессов и архитектур. Присоединившийся к нам Джим поможет ускорить эти преобразования», заявил главный инженер-разработчик и президент группы TSCG Intel

д-р Мерти Рендучинтала [Murthy Renduchintala]. После ухода из Tesla Криса Лэтнера [Chris Lattner], Келлер, помимо своей роли вице-президента по низковольтному оборудованию, отвечал за программу Autopilot autonomous driving: по слухам, Tesla была заинтересована в разработке собственного процессора для поддержки своих амбиций по созданию автопилотируемого автомобиля. Келлер обладает более чем 20-летним опытом разработки микроархитектур на основе x86 и ARM для ПК, серверов, мобильных устройств и транспортных средств в DEC (где он работал над микропроцессором Alpha), AMD, PA Semiconductor и Apple (где он работал над CPU на основе архитектуры ARM A4 и A5). Однако прославился он, когда помог AMD первой преодолеть барьер тактовой частоты в 1 ГГц, опередив в этом рекорде Intel. В 2012 г. он вернулся в AMD из Apple для работы над микроархитектурой Zen, на основе которой с февраля прошлого года выпускается линейка микропроцессоров Ryzen.



Джим Келлер: «В следующем десятилетии мир, управляемый компьютерами, станет совершенно иным. Я рад присоединиться к команде Intel, создающей CPU, GPU, ускорители и другие продукты для ориентированной на большие данные компьютерной эры».

ОТВЕТНЫЙ ХОД

Еще раз про... SSD

Intel выпустила очередной твердотельный накопитель Optane 905P.

Ответом на выпуск Samsung «безумно быстрых» SSD 970-й серии можно считать новейший твердотельный накопитель Optane от Intel: его наиболее яркая характеристика — тоже невероятная скорость. Для последовательных чтения и записи показатели SSD Intel Optane 905P — 2600 МБ/с и 2200 МБ/с соответственно, что очень похоже на параметры SSD других производителей. А вот для произвольных чтения и записи новый Optane обеспечивает 575 тыс. IOPS и 555 тыс. IOPS; для многих ключевых задач будет ощутима разница — хранящиеся на диске данные будут найдены и нужные действия с ними выполнены почти мгновенно после нажатия на клавишу Enter. Optane 905P на 25 тыс IOPS (чтение) и 50 тыс. IOPS (запись) быстрее предыдущего поколения накопителей Intel, Optane 900P, выпущенного менее года назад (особенно стоит «убийца SSD» Optane 800P, в котором задействована память 3D XPoint вместо традиционных для SSD технологий и который по-прежнему слишком дорог для широкого распространения — пусть нумерация моделей не даст вам себя обмануть).

Кроме того, новый SSD имеет удвоенный объем 960 ГБ против 480 ГБ у 900P. Цена в Великобритании за модель объемом 960 ГБ в PCIe-версии составит £830, а за более традиционную версию 2,5", емкостью только 480 ГБ — около £445. Выполненный в виде карты расширения PCIe накопитель имеет RGB-подсветку: видимо, производители до сих пор почему-то убеждены, что пользователям буквально не жить без мерцающих огней вокруг непрозрачного черного корпуса.

Новый SSD не дешев, и отлично подходит для профессионального применения. Для хранения фотографий любимой кошки его применение пока менее обосновано. Но рано или поздно цены на такие изделия опускаются до разумного предела.



Накопители Intel Optane SSD 905P будут доступны в трех вариантах исполнения (на снимке — плата расширения PCI Express x4 емкостью 960 ГБ).

ДИСТРИБУТИВ МЕСЯЦА

Один из «большой тройки»

Вышло традиционное апрельское обновление Ubuntu и его большого семейства.

Состоялся релиз дистрибутива GNU/Linux Ubuntu 18.04 Bionic Beaver и всех его производных (Kubuntu, Ubuntu Budgie, Lubuntu, Ubuntu Studio, Ubuntu Kylin, Ubuntu Mate и Xubuntu). Традиционно для апрельского выпуска в четном году релиз имеет приставку LTS, т.е. обновления для него будут выпускаться пять лет (девять месяцев у обычных релизов). К LTS-релизам разработчики предъявляют повышенные требования по надежности и стабильности, и необкатанных новинок и экспериментальных функций в релизах с долгосрочной поддержкой нет.

Что же мы увидим в этом выпуске?

- Родительская ОС, Ubuntu 18.04 LTS: Gnome 3.28; офисный пакет LibreOffice 6.0; ядро 4.15 и стек Xorg 1.19.6; в графический установщик вошла опция минимализма (только браузер с минимумом утилит).
- Kubuntu 18.04 LTS: KDE Plasma 5.12 и набор приложений KDE Application 17.12.3; VLC вместо Dragonplayer в качестве мультимедиа-проигрывателя по умолчанию; по умолчанию задействован интерфейс управления пакетами Muon как альтернатива Plasma Discover; KTorrent обновлен до 5.1, портированный Qt5/KF5; отсутствует привычный KDE-шникам музыкальный плеер Amarok,

его заменил графический интерфейс к проигрывателю Cantata на Qt5; удален прекративший развитие коммуникационный клиент Telepathy.

- Ubuntu Mate 18.04 LTS: Mate 1.20 с улучшенной поддержкой HiDPI и расширений для X11 в DRI3 (Direct Rendering Infrastructure) и XPresent в оконном менеджере Marco.
- Xubuntu 18.04 LTS: тема Greybird GTK обновлена до версии 3.22.6 со стилями GTK3

для Chrome/Chromium; просмотрщик документов Evince сменил на Atril; архиватор File Roller уступил место Engrampa; вместо Gnome Calculator — Mate Calculator; добавлен модуль PulseAudio для работы с устройствами Bluetooth; обновлены многие приложения, в т.ч. Parole Media Player 1.0.0, Thunar 1.6.14, Xfce Panel 4.12.2.

- Ubuntu Budgie 18.04 LTS (версия с современным, но привычным легковесным рабочим столом от проекта Solus): обновлен Budgie, тема оформления Pocillo для экрана блокировки, улучшена поддержка HiDPI.

К LTS-РЕЛИЗАМ РАЗРАБОТЧИКИ ПРЕДЪЯВЛЯЮТ ПОВЫШЕННЫЕ ТРЕБОВАНИЯ ПО НАДЕЖНОСТИ.

ХРОМБУКИ

Приближая сингулярность

Google предоставляет полную поддержку приложений для Linux на хромбуках.

Гoogle в очередной раз приблизила наступление сингулярности — но пока не в технологическом и не в философском смысле этого слова. Просто теперь GNU/Linux и два его больших ответвления, Chrome OS и Android, становятся немного шире, чем одна операционная система. Для Google Pixelbook это означает реализацию preview-версии поддержки приложений для Linux, и скоро такое смогут и другие хромбуки. К концу года пользователи смогут запускать Linux-терминал, Git, Sublime, Vim и Android Studio; на подходе — спецверсия Android Studio для Chrome OS. Как утверждает Google, удалось добиться «бесшовного» функционирования — пользователь выбирает нужное приложение, независимо от ОС, и оно будет работать в среде Chrome OS. Поддержка приложений Linux означает, что разработчики, наконец, смогут применять хромбуки для разработки для платформ Google, не завися от компьютеров с Windows, Mac или Linux. На хромбуках разработчики

смогут создавать, тестировать и запускать любые Android- или web-приложения для телефонов, планшетов и ноутбуков в своей любимой IDE. Высказывались предположения, что Google готова представить вариант хромбука с двойной загрузкой в Windows, в ответ на обвинения, что цена на Pixelbook как устройство, игнорирующее самую популярную ОС в мире, слишком завышена. Тем не менее, в результате на Pixelbook теперь есть практически всё, чего может пожелать программист, что делает Chrome OS как платформу для кодеров намного привлекательнее. Для прочих пользователей Linux на хромбуках устраняет многие проблемы в Chrome OS/Android, добавляя массу полноценных настольных приложений и игр. Но по умолчанию поддержка приложений Linux в Chrome OS включена не будет: пользователям придется самим включить ее в настройках, поскольку, по мнению представителей Google, «средний пользователь не нуждается в терминале Linux и возможностях впрограмирования».



Первым хромбуком с интегрированной поддержкой приложений для Linux стал Pixelbook.

Источник: distrowatch.com; www.theinquirer.net

СКАЖИ УЧЕБЕ

ДА!



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

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

UnixEducationCenter

СУПЕРКОМПЬЮТЕРЫ

Intel и Nvidia придется потесниться

В Великобритании развернут один из крупнейших в мире комплексов на ARM.

Три кластера суперкомпьютеров в университетах Эдинбурга, Бристоль и Лестера будут насчитывать более 12 тыс. ядер ARM и включать по 64 системы HPE Apollo 70, каждая из которых оснащена двумя 32-ядерными процессорами Cavium ThunderX2 и 128 ГБ ОЗУ, состоящей из 16 модулей DDR4 DIMM с межсоединениями Mellanox InfiniBand; ОС — SUSE Linux Enterprise Server for HPC. Каждый кластер разместится в двух Rack-стойках, его энергопотребление составит 30 кВт. Работы по установке завершатся летом 2018 г.; это будет одна из крупнейших в мире HPC-систем на базе ARM (ее стоимость не раскрывается). ARM стремится к дальнейшему продвижению на рынке HPC, на котором в настоящее время доминируют Intel и Nvidia: в конце минувшего года Cray заявила, что создает первый в мире суперкомпьютер

промышленного уровня с процессорами Cavium ThunderX2 на 64-разрядной архитектуре ARMv8-A. «У нас уже есть 2 национальных сервиса HPC на технологии HPE. Данный проект будет нашим первым крупномасштабным суперкомпьютером на ARM. В случае успеха процессо-

ARM СТРЕМИТСЯ К ПРОДВИЖЕНИЮ НА РЫНКЕ ВЫСОКОСКОРОСТНЫХ ВЫЧИСЛЕНИЙ.

ров ARM в качестве суперкомпьютерной технологии потребуются создание серьезной экосистемы ПО, и EPCC готов портировать многие из ключевых научных приложений UK для нашей системы HPE Apollo 70», говорит директор EPCC университета Эдинбурга Марк Парсонс [Mark Parsons].

APU AMD

Экономим электроэнергию

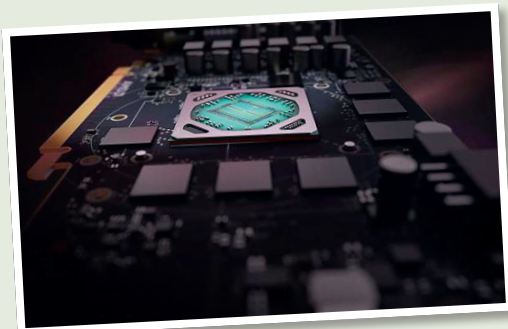
Рассеяние мощности новых процессоров AMD — всего 35 Вт.

AMD готовит выпуск экономичных чипов Ryzen 3 2200GE и Ryzen 5 2400GE, с фирменными GPU Vega. При одинаковом с предшественниками, Ryzen 3 2200G и Ryzen 5 2400G, количестве ядер и потоков, требование по теплоотводу (TDP) у новинок — 35 Вт. Тактовая частота чипов с маркировкой “GE” ниже, чем у более энергоемких “G”, TDP которых 65 Вт, но CPU и GPU практически идентичны. Соответственно, энергопотребление и расход батарей устройств на новых чипах будут существенно меньше. Пока AMD представила чипы “GE” в формате AM4, однако скоро, несомненно, появится версия для ноутбуков, где важны энергоэффективность и увеличение срока службы батарей.

Ryzen 3 2200GE имеет 4 ядра и 4 потока, но многопоточная обработка данных (SMT) не поддерживает; базовая частота 3,2 ГГц, ускоряемая до 3,6 ГГц (соответствующие показатели 2200G — 3,5 и 3,7 ГГц), частота интегрированного GPU Vega 8 — 1100 МГц. У Ryzen 5 2400GE также 4 ядра, но есть поддержка SMT, 8 по-

токов, базовая частота 3,2 ГГц, ускоряемая до 3,8 ГГц (у 2400G — 3,6 ГГц и 3,9 ГГц соответственно), частота интегрированного GPU Vega 11 — 1250 МГц.

Рекомендованных цен на новинки компания пока не называет (для сравнения, цена Ryzen 3 2200G — \$99, Ryzen 5 2400G — \$169). **LXF**



AMD продолжает продвигать комбинацию CPU Ryzen и GPU Vega в едином чипе.

Короткой строкой

➤ Huawei — первый поставщик, успешно прошедший тесты ядра сети для неавтономной архитектуры 5G (NSA).
Источник: www.huawei.com

➤ Компания МЦСТ (автор процессоров Эльбрус) выдала инженерные образцы R-2000 (процессор класса RISC, на 64-разрядной открытой архитектуре SPARC v.9, доработанной компанией).
Источник: www.cnews.ru

➤ Единственный разработчик дистрибутива Korora объявил о приостановке разработки.
Источник: kororaproject.org

➤ В Chrome 68 все сайты с HTTP получат индикатор небезопасности; для HTTPS в 69-й версии уберут надпись “Secure”, а в 70-й — значок с замком.
Источник: blog.chromium.org

➤ Начиная с релиза 18.10, в Ubuntu официально принят рабочий стол LXQt.
Источник: ubuntu.me

➤ В Solaris 11.3 SRU 32 — новые системные вызовы `recvmsg()` и `sendmsg()`; режим сетевых сокетов `SO_REUSEPORT`; приложения *Explorer 18.2* и *SunVTS 8.2.2*; команда `netstat -P udp`.
Источник: www.opennet.ru

➤ МТС будет продавать в РФ смартфон Moto Z2 Force на технологии LAA, с передачей данных до 1 Гбит/с за счет агрегации LTE и Wi-Fi. Такая сеть уже есть в Уфе.
Источник: www.cnews.ru

➤ «Лаборатория Касперского» сочла эксплойты для MS Office одной из самых распространенных угроз в I квартале 2018 г.
Источник: www.kaspersky.ru

➤ На основе Минкомсвязи создано Министерство цифрового развития и связи.
Источник: www.cnews.ru

➤ Раскрыты данные об уязвимости CVE-2018-8897 64-разрядных процессоров AMD и Intel, для Linux, FreeBSD, Windows, macOS, Xen, KVM и VMware.
Источник: www.opennet.ru

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!

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



Обзоры



Алексей Федорчук
Тэг <сарказм>
по умолчанию,
смайлики по вкусу.

» BIONIC SERVER КАК ДЕСКТОП

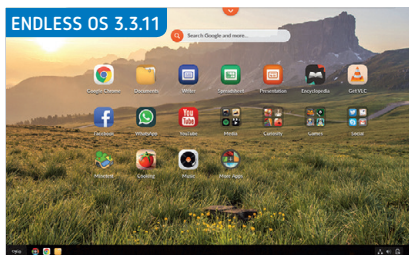
О системе Cintu немало говорилось на этих страницах. И всегда подчеркивалось, что в качестве основы для нее использовался установочный диск Ubuntu **mini.iso** — образ объемом в несколько десятков мегабайт, включающий ядро Linux, средства запуска инсталлятора (так называемого «альтернативного», работающего в текстовом режиме) и настройки сети — всё остальное, и в наисвежайшем виде, получается из официальных репозиториях проекта. Метода была отработана, проста, надежна.

Но случилась незадача: на стадии тестирования 18.04 в установщике **mini.iso** безнадежно поломали консольную русификацию — похоже, это становится традицией перед выходом очередного LTS-релиза. Само по себе — не страшно, на стадии инсталляции можно и вражью мову стерпеть. Но создавало некоторые шероховатости уже после установки. Впрочем, тоже поправимые — но нельзя ли обойтись без них? А потому появилась мысль: в качестве базиса для Cintu использовать Ubuntu Server. Тем более, что в нем появился новый инсталлятор, тоже текстовый, но вполне современный на вид («альтернативный» инсталлятор Ubuntu не менялся лет 14).

Оказалось, что инсталлятор очень даже годный, весьма простой и при этом вполне функциональный. Правда, неизбежны затраты дискового пространства на избыточные цели — но это решаемо. Его бы в **mini.iso** прикрутить, с опцией установки только метапакета *ubuntu-minimal* — цены для наших целей бы не было. Так что если в «альтернативном» инсталляторе консольную кириллицу к релизу не вылечат, можно будет и его принять за основу.
alv@posix.ru

Сегодня мы рассматриваем:

- Univention Corporate Server 4.3** 14
Ориентированный на малые и средние предприятия дистрибутив может похвастаться сравнительной простотой установки. Преодолев бывшие недостатки, он заметно похорошел.
- Endless OS 3.3.11** 15
Данный дистрибутив не похож на любой популярный настольный вариант: он целиком и полностью использует Flatpak. Беда в том, что отнюдь не все приложения перешли на этот формат..
- AMD Ryzen 5 2400G** 16
Лучший из когда-либо выпускавшихся APU AMD непринужденно побивает
- по очкам Ryzen 5 1500X и прочие процессоры в своей категории. Вполне достоин вашего ПК.
- Seagate IronWolf 12TB** 18
Твердотельные накопители, конечно, работают с огромной скоростью, но влетят в копейку. Поэтому жесткие диски продолжают себе крутиться — и в общем, свое дело делают.
- Railway Empire** 19
Если в детстве вы не наигрались в железную дорогу, вот вам отличный шанс восполнить этот пробел — на более сложном уровне живучей капиталистической мечты.



За символ бесконечности, видимо, выбраны бескрайние поля и далекие горы.



Процессор со встроенной графикой нового поколения – первенец AMD.

» СРАВНЕНИЕ: ДИСТРИБУТИВЫ NAS C. 22

Наши домашние архивы стремительно прирастают терабайтами: тут и фотки (в цифровой век они плодятся как кролики), и видео, и музыка. Гроздьями дисков уже не отбиться...

Univention Server 4.3

Хорошо продуманный интерфейс управления сумеешь оценить, только сражаясь с простудой, как, чихая, обнаружил **Шашанк Шарма**...

ВКРАТЦЕ

Основанный на Debian 9, Univention предназначен для малых и средних предприятий. Его веб-интерфейс управления может использоваться для установки дополнительного ПО и настройки различных элементов дистрибутива. При каждой возможности дистрибутив использует контейнеры Docker, и теперь поддерживается более 90 различных модулей, таких как принтер и файловый сервер, веб-прокси, антивирус и др. Также см. Zentyal, ClearOS и NethServer.

Как и большинство других аналогов, Univention Corporate Server ориентирован на малые и средние предприятия. Правда, он помогает пользователям установить рабочий стол KDE, но чересчур специфичен для случайного или обыкновенного домашнего пользователя.

Дистрибутив может похвастаться довольно несложным процессом установки. Выбор Debian в качестве базы делает Univention невероятно безопасным, и дополнительные компоненты, такие как антивирус и брандмауэр, устанавливаются легко.

В дополнение к 1,3-ГБ устанавливаемой ISO-образу, проект также предоставляет онлайн-демо. Если хотите пропустить установку и сразу перейти к Univention, проект также обеспечивает выбор предустановленных образов как для VMWare, так и для VirtualBox.

Установка позволяет настроить схему разбиения и даже выбрать пакет установки. Например, можно установить рабочий стол KDE и Fetchmail, CUPS и другие важные компоненты.

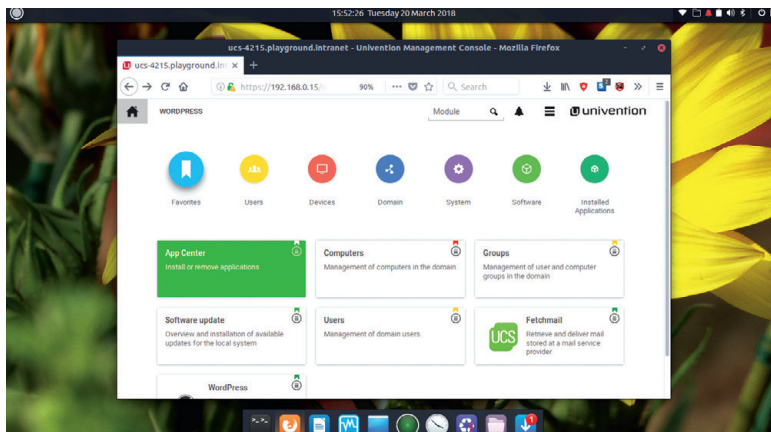
Держать в уме порталы

В отличие от других серверных дистрибутивов, Univention предоставляет строго ограниченную учетную запись администратора, и рекомендует входить в систему как Администратор, используя пароль привилегированного пользователя, который вы настроили во время установки.

При первом входе в систему вы попадаете на страницу портала, которая предоставляет полный обзор всех доступных сервисов в вашей инсталляции Univention и дает ссылку на Univention Management Console (UMC). Можно считать UMC центральным узлом настройки веб-интерфейса. Здесь есть ряд ярлыков в верхней части экрана, такие как Users, Devices, Domain, System и Software. Интерфейс интуитивно довольно понятен и удобен в навигации. Большую часть параметров настройки вы найдете припрятанную под Системной меткой на «приборной панели» Dashboard.

При нажатии на Software интерфейс покажет список всех установленных и доступных программ. Можно щелкнуть по любой программе, например, NextCloud, и вам предоставят данные о ее функциях и лицензиях. Внутри уникальных контейнеров Docker установлено кое-какое ПО, например, NextCloud и WordPress, и перед началом установки дистрибутив предупреждает об этом. К сожалению, Univention не сообщает общий размер загрузки при установке пакетов.

В последней версии видно, что на веб-портале проделано много работы. Теперь можно менять внешний вид разных страниц и перемещать элементы путем перетаскивания. Благодаря расширенному пользовательскому управлению, можно настраивать содержимое, доступное для видения пользователей, основанное на том, в каких группах они состоят, что предпочитают и других параметров.



Обновления безопасности для каждой версии доступны в течение 6 месяцев после выпуска – коммерческие предприятия это оценят.

Еще одной новой особенностью выпуска 4.3 является использование Kerberos для обеспечения единого входа в систему. Теперь, когда пользователи входят в Univention, они немедленно получают доступ к учетным записям Office 365 или NextCloud, для этого не придется отдельно логиниться в системе.

Типичный бизнес-сервер, естественно, не обходится без кривой обучения; к счастью, Univention предоставляет не только краткое руководство для пользователей, а еще и форумы для подсобединения к поддержке сообщества пользователей. Там вы также найдете массу подробных руководств и скриншотов по разным темам. Если есть интерес к платной поддержке, то проект также предоставляет различные варианты подписок с поддержкой предприятия и гарантией техобслуживания за плату €290 в год.

Месяцев 9 назад, в выпуске LXF227, мы провели обзор дистрибутивов — бизнес-серверов, в котором Univention Corporate Server 4.2 выступил довольно плохо, несмотря на сильное финиширование почти во всех тестах. И в самой первой после этого версии дистрибутив разобрался почти со всеми основными недостатками, которые стоили ему призового места. **LXF**

ВЕРДИКТ

РАЗРАБОТЧИК: Univention GmbH
САЙТ: www.univention.com
ЛИЦЕНЗИЯ: AGPL

ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	9/10
УДОБСТВО В РАБОТЕ	9/10
ОПРАВДАНОСТЬ ЦЕНЫ	9/10

Производительность резко улучшилась, а контейнер Docker и функции, например, единый вход в систему, очень радуют.

➤ **Рейтинг 9/10**

Endless OS 3.3.11

Шашанк Шарма озадачен: сам по себе дистрибутив бесконечно впечатляет, но, при его огромном 15-ГБ объеме загрузки, не просто ли это куча байтов?

ВКРАТЦЕ

Дистрибутив на базе Debian и с модифицированным рабочим столом Gnome поставляется в двух вариантах. 2-ГБ ISO — полностью функциональный дистрибутив, но ему не хватает некоторых полезных приложений. Вариант 15 ГБ поставляется с сотней предустановленных приложений. Его индивидуальный рабочий стол Gnome предлагает удобный интерфейс. Управление пакетами фокусируется на Flatpak, этим он отличается от аналогов вроде Elementary OS и Ubuntu.

Благодаря индивидуально настроенному рабочему столу Gnome, Endless [англ. бесконечный] OS не похож на любой популярный настольный дистрибутив. Хотя Endless OS и основан на Debian, но его подход к управлению ПО и обновлениям системы значительно отличается. Фактически, *apt* или любой другой пакет управления системой отпали, поскольку для предоставления приложений пользователям дистрибутив использует Flatpak.

Удивительно, что в отличие от других дистрибутивов Linux, Endless OS не допускает двойной загрузки с другим экземпляром Linux, но может сосуществовать с Windows. Это и другие отклонения от обычной практики Linux превращают дистрибутив в легко осваиваемую альтернативу для нетехнических и неопытных пользователей.

Первоначально доступный только на бюджетных машинах от Endless, дистрибутив теперь поставляется в двух вариантах. Если вам нужен удобный оффлайн-доступ к Wikipedia и еще тысячам руководств пользователя и инструкциям, размещенных на WikiHow, а также множество других ресурсов, то загрузите полную версию, которая весит невероятные 15 ГБ. Версия-мастодонт также может похвастаться 100 приложениями по категориям, например, играми, мультимедиа и различными учебными и обучающими инструментами для детей, например, Scratch. В отличие от других настольных дистрибутивов, которые спокойно можно установить на дисках емкостью 10 ГБ, да еще и куча места останется, для Endless OS рекомендуется не менее 32 ГБ, а установка по умолчанию займет 20 ГБ дискового пространства.

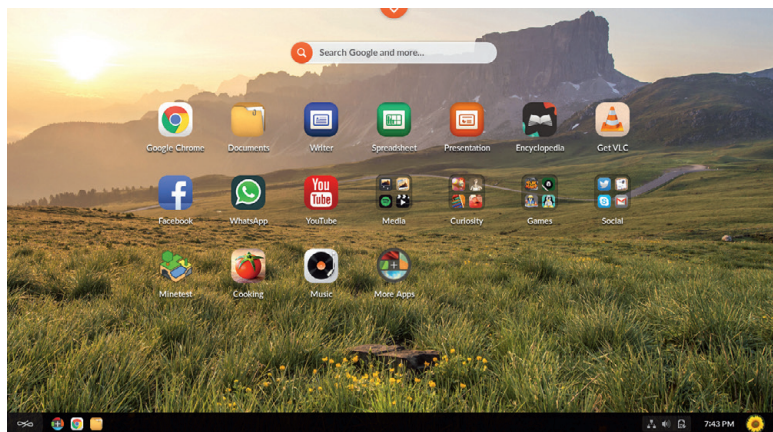
Бесконечные различия

В дистрибутиве есть возможность запуска с USB-накопителя. Если вы не собираетесь выполнять двойную загрузку с Windows, Endless OS займет целый диск, и установщик потребует от пользователя лишь небольшого участия. Вам не разрешено даже предоставлять для дистрибутива пользовательскую раскладку, например, отдельный/домашний раздел.

Большая часть настроек, в частности, настройка учеток социальных сетей и прочего, поручены пост-установочному мастеру конфигурации. По окончании вы попадете на почти неизвестаемый рабочий стол Gnome. Его дизайн склоняется к устройствам с сенсорным экраном, что очевидно по отсутствию меню запуска приложений, стандартного для большинства сред рабочего стола.

Вы можете использовать глобальную панель поиска для поиска приложений и даже для поиска в Интернет. Многие приложения, такие как терминал, на рабочем столе не отображаются. Чтобы получить доступ к этим и большинству из других 100 приложений, нужно запустить *App Center*. Если ваши любимые повседневные приложения еще не на рабочем столе, можно открыть их в *App Center* и нажать кнопку *Add to Desktop*.

Хотя в дистрибутиве имеется медиа-плеер и поддерживается масса аудиоформатов, из коробки большинство популярных



Особый рабочий стол Gnome предлагает несколько рядов приложений. Нажатие Curiosity или Encyclopedia прояснит причину огромного объема его загрузки.

проприетарных форматов работать не будет. Но вы можете приобрести кодеки, чтобы позволить Endless OS воспроизводить MP4 и т.п.

Автоматические обновления дистрибутива по умолчанию не будут проблемой для пользователей с высокоскоростным Интернет-доступом без FUP [fair use policy, ограничение скорости интернет-провайдером]. Использование OSTree для установки атомарных изменений, которые не влияют на текущее состояние дистрибутива, является еще одним отходом от корней Debian. Когда дистрибутив больше всего полагается на Flatpak для предоставления приложений, не удивляйтесь, если в Центре приложений вы не сможете найти кое-какие из самых популярных.

Полезность такого дистрибутива, как Endless OS, нельзя отрицать. Тем более что он может быть отличным инструментом обучения, а также защиты юношества от неприятных частей Интернета. К сожалению, он требует больше ресурсов, чем его популярные коллеги — Ubuntu, Fedora и другие, и порой его медлительность ощущается даже при 4 ГБ оперативной памяти. **LXF**

ВЕРДИКТ

РАЗРАБОТЧИК: Endless
САЙТ: www.endlessos.com
ЛИЦЕНЗИЯ: Различные открытые

ФУНКЦИОНАЛЬНОСТЬ	7/10
ПРОИЗВОДИТЕЛЬНОСТЬ	7/10
УДОБСТВО В РАБОТЕ	10/10
ДОКУМЕНТАЦИЯ	8/10

Зависимость от Flatpak делает дистрибутив негибким и неподходящим для тех, кто любит экспериментировать с установкой Linux.

» Рейтинг 8/10

AMD Ryzen 5 2400G

Скорость встроенной графики нового поколения наконец попала в процессор, достойный вашей основной настольного ПК. **Джарред Уолтон** впечатлен.

СПЕЦИФИКАЦИЯ

- Разъем AM4
- Разрядность 64 бит
- Технологический процесс 14 нм
- Ядра 4
- Потоки восемь
- Тактовая частота 3,6 ГГц (3,9 ГГц турбо)
- Кэш L1 384 кБ, L2 2 МБ, L3 4 МБ
- Память DDR4, два канала, поддержка ECC
- Тепловыделение 65 Вт
- PCIe 3.0 8 линий
- GPU Radeon RX Vega 11
- Тактовая частота GPU 1,25 МГц

После яркого прорыва процессоров Ryzen в 2017 г. многие возлагали большие надежды на будущие APU Ryzen (что со встроенной графикой), и, наконец, AMD выпустила первые модели. Вечная проблема с интегрированной графикой заключается в попытке совместить хороший процессор с достойным графическим процессором, сохраняя при этом цену, мощность и другие аспекты. Это лучший APU AMD, который когда-либо выпускался, и он эффективно уничтожает Ryzen 5 1500X и более слабые процессоры.

По части самого процессора, первая партия процессоров Ryzen использовала одну структуру CCX (CPU Complex), а Threadripper дошел до включения двух таких чипов в один корпус. Ryzen CCX состоит из четырех ядер процессора, у каждого из которых 512 КБ кэша L2 и общий кэш L3 объемом 8 МБ. В предыдущих четырехъядерных реализациях (Ryzen 3 1200/1300X и Ryzen 5 1400/1500X) каждый CCX в итоге два ядра отключал, и в зависимости от модели половина кэша L3 тоже может быть отключена. У APU Ryzen есть только один CCX и максимум 4 МБ кэша L3. Это упрощает некоторые аспекты многоядерной работы — нет задержек между CCX, о которых надо беспокоиться; однако уменьшенный размер кэша L3 иногда может сказываться на производительности.

Вместо второго CCX AMD включил небольшое приятное графическое решение. Ryzen 5 2400G называет его Vega 11, являющегося ядром Vega с 11 включенными CU (вычислительные блоки), каждый из которых включает 64 потоковых процессора (они же — ядра GPU). Число одиннадцать выглядит странно, так что где-то может скрываться еще один отключенный CU; но AMD это не комментирует.

Как и в большинстве интегрированных решений, здесь нет выделенной графической памяти, что влияет на производительность. Но Vega 11 привносит все другие новые функции архитектуры Vega, такие, как быстрая математика (иначе, FP16), Draw Stream Binning Rasterizer, переработанный механизм геометрии и многое другое. Интересно, что Intel собирается обойти это в своих будущих чипах Core + Vega M, включив 4-ГБ HBM2, который должен оказаться значительно быстрее, но и намного дороже, и только для мобильных решений.

Разблокирован и загружен

Как и все процессоры AMD Ryzen, Ryzen 5 2400G поставляется полностью разблокированным — как со стороны CPU, так и со стороны GPU. Не будем пока раскрывать показатели производительности, но 2400G по существу звучит похоронным звоном для уже имеющегося Ryzen 5 1500X. Он официально оценен ниже, его тактовые частоты чуть повыше, но имеет половину кэша L3, на 12 меньше линий PCIe и такой же предел разгона. Оба чипа работают в гнезде AM4, так что теперь почти нет причин покупать четырехъядерное решение Ryzen 5 — или процессоры Ryzen 3, если на то пошло.

Испытательная система, поставляемая AMD, чрезвычайно компактна и демонстрирует тип сверхмалого решения mini-ITX, которое идеально подходит для корпуса домашнего кинотеатра. При простое вся система потребляет всего 21 Вт, а при полной нагрузке — доходит до 90 Вт. Вам не понадобится сумасшедшая система охлаждения, чтобы поддерживать работу в приемлемом диапазоне температур.

Что вы получаете от 704 потоковых процессоров с тактовой частотой 1250 МГц, общим системным ОЗУ с четырехъядерным восьмипоточным процессором Ryzen? Не следует удивляться, что производительность графики значительно отстает от RX 560 4 ГБ, у которой аналогичные базовые тактовые частоты, но 1024 ядра и выделенная полоса пропускания GDDR5 в 112 ГБ/с. Производительность в целом будет по-прежнему ниже, чем даже у RX 550 (512 ядер и 112 ГБ/с VRAM) благодаря памяти GDDR5.

» РЕЗУЛЬТАТЫ ИГРОВЫХ ТЕСТОВ

Tomb Raider (2013 г.)

Кадров	Мин.	Макс.	Среднее
«Нормально»	46	73	61
На пределе	23	36	30

Heaven

Кадров	Мин.	Макс.	Среднее
По умолчанию без AA	8	57	29
По умолчанию 4xAA	12	36	21,4

Dirt Rally

Кадров	Мин.	Макс.	Среднее
По умолчанию без AA	49	70	59
По умолчанию 4xAA	28	52	37
Ультра	34	56	41

*все игры тестировались при разрешении 1920×1080

Geekbench

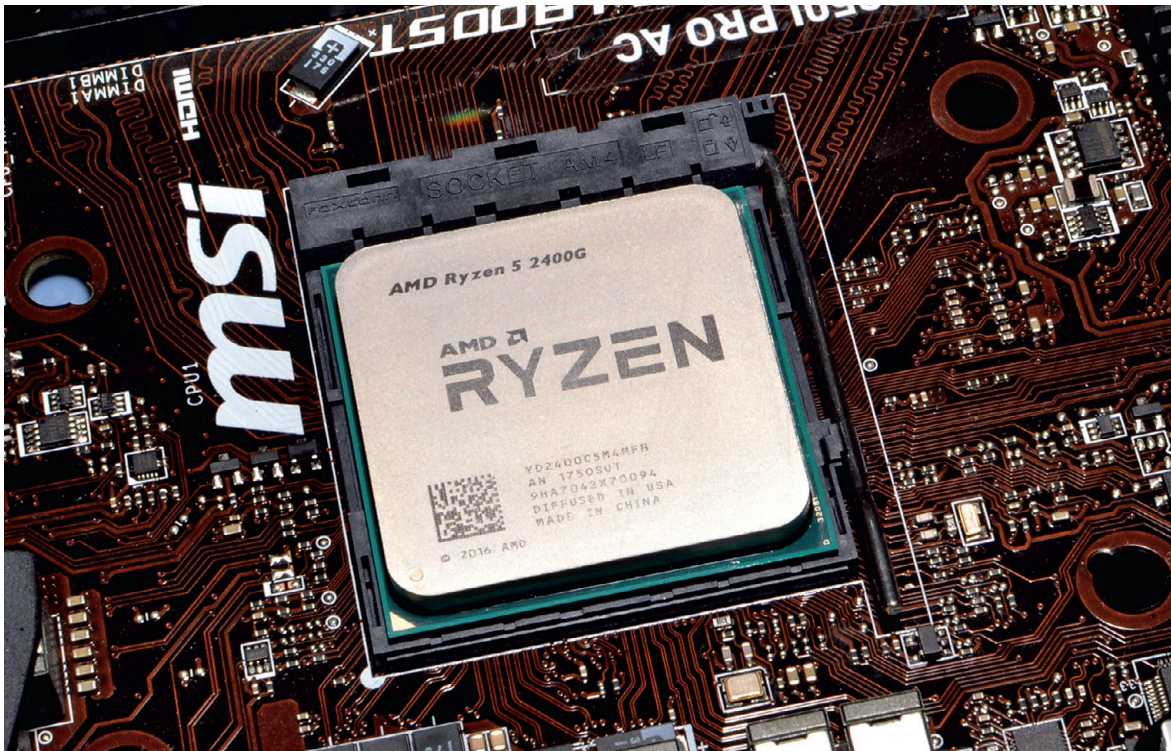
Одноядерный	4792
Многоядерный	14688

<https://browser.geekbench.com/v4/cpu/7272159>



Коробочная розничная версия AMD Ryzen 5 2400g поставляется с низкопрофильным маломощным кулером Wraith Stealth.

На сегодняшний день AMD Ryzen 5 2400G имеет лучшую интегрированную графику.



Перейдем к неигровым тестам [benchmark]: многие тесты показывают виртуальную ничью между 2400G и 1500X. В целом, 2400G и 1500X фактически одинаковы. Однопоточный Cinebench, например, предпочитает 2400G, возможно, из-за отсутствия задержек между CCX или, возможно, просто из-за другой прошивки и материнской платы. Новые APU убрали потребность в четырехъядерных процессорах Ryzen: APU имеют аналогичные характеристики и работают примерно одинаково, и включают графику в виде бонуса.

Обзор видео

Когда производительность ЦП не так важна, многим пользователям не нужна подобная лошадиная сила. Отличным примером является воспроизведение видео. Мы транслировали разнообразный HD-контент на 2400G, всё без нареканий и с чрезвычайно низким энергопотреблением: 28–30 Вт для контента 1080p, 35–40 Вт для потоковой передачи 4K, без каких-либо прерываний. Технически, Ryzen также поддерживает злую магию DRM, необходимую для потоковой передачи Netflix или Amazon 4K, но будет ли разрешено Linux получать такие потоки — еще тот вопрос.

Неизбежно, что в какой-то момент AMD создаст APU, похожий на будущий процессор Intel восьмого поколения с графикой Vega M, что означает больше графических ядер и какое-то выделение HBM2 VRAM. Это может произойти в 2019 г., или дождется той поры, когда процессоры 7 нм станут обычным делом; но более высокие уровни интеграции неизбежны. Сейчас мы вынуждены мириться с совместным использованием, а графика и процессоры не очень хорошо решают, кто получает наибольшую пропускную способность памяти — точнее, графические ядра, как правило, забирают всю полосу, какую вы можете дать, по крайней мере, до диапазона больше 100 ГБ/с, а системная память просто недостаточно быстра. Мы можем наблюдать резкое падение производительности при переключении с быстрой памяти DDR4-3200 на более типичную бюджетную DDR4-2400. Это равнозначно потере примерно 20% производительности.

Имейте в виду, что большинство материнских плат AM4 должны поддерживать новые чипы, а более старые платы потребуют

обновления BIOS, для чего может понадобиться специальная карта. У AMD есть страница поддержки, описывающая проблему.

Поддержка Linux выглядит хорошей. Вам нужно ядро Linux 4.15 для поддержки нового APU из коробки. Рекомендуется использовать ядро 4.16, так как оно содержит исправления ошибок и оптимизацию. Для многих это будет означать Ubuntu 18.04 LTS, что и неплохо. Мы тестировали дневную сборку Ubuntu 18.04 Alpha 2 с ядром 4.15, и она работала безупречно из коробки; нет загрузки драйверов, нет обновлений Mesa. В двух играх, которые мы пробовали, были проблемы: *Metro Last Light* повесила систему, а *Dying Light* потребовала обновления Mesa (это может быть исправлено к моменту, когда вы это читаете). С другой стороны, результаты получены со стандартной установкой Ubuntu.

Кто будет покупать Ryzen 5 2400G? Бюджетные игроки, ищущие временную замену выделенной графической карте, должны заинтересоваться. Больше всего выигрывают домашние кинотеатры. Поместите его в корпус мини-ИТХ, с хорошим сетевым подключением и большой емкостью сетевого хранилища, и вы готовы к полноценной игре на 1080p, к тому же бесплатно. **LXF**

ВЕРДИКТ

ПРОИЗВОДИТЕЛЬ: AMD

САЙТ: www.amd.com

ЦЕНА: £ 140

ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
УДОБСТВО В РАБОТЕ	9/10
ОПРАВДАНОСТЬ ЦЕНЫ	9/10

Самая быстрая интегрированная графика за приличную цену; отлично работает как НТРС, или для обычных игровых требований.

» **Рейтинг 9/10**

Seagate IronWolf 12TB

Кто испугается большого страшного «Железного Волка» в наше время сверхъёмких и сверхбыстрых SSD? — вопрошает **Джереми Лэйрд**.

СПЕЦИФИКАЦИЯ

- **Емкость 12 ТБ** (есть модели до 1 ТБ)
- **Интерфейс SATA** 6 Гб/с
- **Протокол AHCI**
- **Скорость** 7200 об/мин
- **Макс. скорость передачи данных** 210 МБ/с
- **Объем буфера** 256 МБ
- **Пластины** 8×1,5 ТБ
- **Средняя нагрузка на отказ** Миллион часов
- **Гарантия** Три года

В наши дни скоростной накопитель видится твердотельным. SSD не только обеспечивают самую высокую производительность, но и предлагают самые большие емкости, достигающие 30 ТБ — однако недешево.

Тем не менее, есть один показатель, по которому всё еще доминируют прежние магнитные жесткие диски: емкость на доллар. Если вам нужна максимальная плотность хранения по минимально возможной цене, тут не обойтись без вращающихся магнитных пластин. Имея это в виду, не удивительно обнаружить, что обычная технология жестких дисков продолжает развиваться, несмотря на понижение до простого ширпотреба. Чуть более года назад самые большие типичные жесткие диски тянули на 10 ТБ. Сейчас мы рассматриваем наш первый 12-ТБ диск Seagate IronWolf.

Двойное волжье удовольствие

Seagate предлагает две версии IronWolf: vanilla и Pro. Диск vanilla предназначен для домашнего пользования, SoHo и приложений SMB, а версия Pro предназначена для полностью коммерческих и корпоративных пользователей. Оба набиты функциями и технологиями. Они включают технологию Seagate AgileArray, которая использует балансировку в двух плоскостях, оптимизацию RAID и улучшенное управление питанием. Затем есть ослабление вращательной вибрации (RV), в том числе несколько датчиков RV, помогающих поддерживать производительность в многодисковых корпусах NAS.

В варианте 12 ТБ IronWolf имеет в общей сложности восемь пластин, то есть 1,5 ТБ на пластину. Поэтому его 3,5-дюймовый корпус переполнен. Обычных углублений на нижней части диска не видно. Обобщая основные характеристики, скорость вращения шпинделя — 7200 об/мин и большой объем буфера в 256 МБ, работа через стандартный интерфейс SATA 6 Гб/с.

В целом, эта 12-ТБ модель рассчитана на 180 ТБ трафика данных в год, имеет MTBF (средняя наработка на отказ) в один миллион часов и трехлетнюю ограниченную гарантию. Для справки, IronWolf Pro оптимизирован для 300 ТБ в год, имеет MTBF в 1,2 миллиона часов и пятилетнюю гарантию. Всё это вдобавок к чертовски большой емкости по довольно разумной цене. Разумеется, диск на 3 или 4 ТБ может предложить несколько большую емкость на один доллар, но 12-ТБ IronWolf вполне

Малый бизнес с ограниченным бюджетом заинтересуется Seagate IronWolf.



конкурентоспособен в этом отношении и предлагает непревзойденную плотность данных.

Так что 12-ТБ IronWolf выглядит неплохо на бумаге и предлагает приятное соотношение емкость — цена. А как он работает? По стандартам обычного магнитного привода он очень быстр. Вы видите последовательные чтение и запись в области 260–270 МБ/с. В скорости произвольного доступа магнитные технологии 4К действительно показывают свой возраст по сравнению с твердотельным хранилищем. Чтение 4К около 1 МБ/с и запись менее 3 МБ/с были бы совершенно катастрофическими для SSD, но для этого типа дисков они вполне конкурентоспособны.

Суть в том, что вы не получите производительность даже бюджетного SSD. Оно и так понятно. Однако, если вам нужна большая плотность хранения по разумной цене, IronWolf дает это и обеспечивает отличную производительность для данного типа дисков. **LXF**

ВЕРДИКТ

ПРОИЗВОДИТЕЛЬ: Seagate
САЙТ: www.seagate.com/ironwolf/
ЦЕНА: £ 329 (12 ТБ), £ 262 (10 ТБ)

ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	6/10
УДОБСТВО В РАБОТЕ	9/10
ОПРАВДАНОСТЬ ЦЕНЫ	10/10

Выдающаяся оправданность цены для своей емкости и производительности, насколько можно ожидать от вращающейся железяки. И прекрасная гарантия.

» **Рейтинг 9/10**

» СРАВНИМ СКОРОСТИ ДИСКОВ

Модель жесткого диска	Seagate IronWolf 12 ТБ	HGST Ultrastar He10 10 ТБ
30-ГБ внутреннее копирование (с)	267	264
Последовательное чтение (МБ/с)	266	246
Последовательная запись (МБ/с)	263	248
4К чтение (МБ/с)	1,16	0,71
4К запись (МБ/с)	2,95	5,97

Railway Empire

Постижение сложности строительства трансконтинентальной железной дороги, сделало **Лейфа Джонсона** самым счастливым управленцем...

СПЕЦИФИКАЦИИ

Минимум

- **ОС** Ubuntu 16.0.4 LTS, 64-бит, SteamOS
- **CPU** Intel Core i5 750, AMD Phenom II X4
- **ОЗУ** 4 Гб
- **Графика** Nvidia GeForce GTX 460, AMD Radeon HD 5870, 2 Гб Vram
- **На диске** 7 Гб

Рекомендуем

- **CPU** Intel Core i5 2400, AMD FX 4100
- **ОЗУ** 4 Гб
- **Графика** Nvidia GeForce GTX 680, AMD Radeon HD 7970, 2 Гб Vram
- **На диске** 7 Гб

В августе 1867 г. банда индейцев-шайенов из штата Небраска попыталась остановить железные дороги, которые несли нежелательные перемены в их родные края. Привязывая толстую палку к путям, они ликовали, когда дрезина сходила с рельс; это побуждало их зарезать всех пассажиров, кроме одного.

В *Railway Empire* [Железнодорожная империя] нет ничего подобного, хотя кампания тоже начинается в 1860-х гг. на плоском и приятном для обучения захватах плато Великих равнин. Сама по себе карта выполнена географически точно, с нежной рябью равнин, перетекающих в гребни Скалистых гор; но эта история сродни очень общему описанию подлинной вещи. Она задает направление как для кампании, так и для сценария — который ставит перед вами задачи, связанные с восстановлением железнодорожной сети Юга после Гражданской войны — но не стоит отправляться в *Railway Empire*, ожидая многому научиться.

Вас проведет по учебному материалу Томас Кларк Дюрант [Thomas Clark Durant], один из важных шишек Объединенной Тихоокеанской железной дороги, завершающей восточную половину Трансконтинентальной железной дороги, но сама карта усыпана городами вроде Майлс-Сити [Miles City], Монтана, и Гранд-Джанкшен [Grand Junction], штат Колорадо, которых в 1860-х гг. еще не существовало. Грозовые тучи иногда затемняют части карты, но нет никаких признаков стада буйволов, которые были вытеснены почти до полного исчезновения — частично, чтобы расчистить путь железным коням. При всей развлекательности *Railway Empire*, она часто предстает как живучая капиталистическая мечта, свободная от всех неудобств прямой конкуренции, которая могла бы помешать получению прибыли.

Из пункта А в пункт В через С-ville

Railway Empire неплохо демонстрирует сложность железной дороги. Рассмотрите фактический акт прокладки путей; самый короткий маршрут между двумя городами может быть самым быстрым, но вы лишаете себя возможности собирать внедорожные товары, например, пшеницу и скот, по пути из Омахи в Денвер.

Идея тут в том, чтобы расположить рядом с этими фермами небольшие станции, упростив перевозку сельхозпродукции в города, которые в ней нуждаются.

Railway Empire удерживает внимание, ориентируясь на интересы



Railway Empire позволяет вам своей рукой соединить уголки Американского континента, умалчивая при этом о влиянии ваших железных коней на экологию.

наблюдателя. И хотя вы нередко имеете дело с проблемой прокладки путей через горы, вас больше занимает поиск денег на это.

Кампания рассчитана на то, чтобы облегчить вам задачу в *Railway Empire*, открывая главу с простых контуров Великих равнин в 1860-х гг., а затем возвращается к малоразвитым 1830-м гг., чтобы противопоставить вас конкурентам на Восточном побережье. Затем установка переходит к логистике гражданской войны, затем к построению моста через реку Миссисипи и, наконец, к западной половине Трансконтинентальной железной дороги, как она прорезает опасный путь через горы Дальнего Запада Америки.

Для более крутого эффекта мы бы предпочли подлинную историю, но впечатляющий дизайн *Railway Empire* особо хорош во Free Mode [Свободный режим], позволяющем вам построить собственные линии без конкретного назначения. XIX век увидел Соединенные Штаты, опьяненные потенциалом потрясающе эффективной новой транспортной технологии. Красота *Railway Empire* — в постоянном воодушевлении от того, что рельсы очерчивают некогда пустынные пространства прерий. **LXF**

Там, где есть железные дороги, обязательно пойдут бизнес.



ВЕРДИКТ

РАЗРАБОТЧИК: Gaming Minds Studios

САЙТ: www.kalypsomedia.com

ЦЕНА: £40

ИГРОВОЙ ПРОЦЕСС	8/10
ГРАФИКА	8/10
УВЛЕКАТЕЛЬНОСТЬ	6/10
ОПРАВДАНОСТЬ ЦЕНЫ	6/10

Добротный симулятор поезда; немного ограничен и недолог, больше для случайных геймеров, но приглянется любителям железных дорог.

» **Рейтинг 7/10**

Мобильные новости

СМЕНА ПОКОЛЕНИЙ

О спорт, ты... интеграция!

Garmin представила платформу Connect IQ 3.0.

На прошедшем в Канзасе втором ежегодном мероприятии Connect IQ Developer Summit американская компания Garmin представила новое поколение приложений для носимых и ориентированных на спорт устройств — Connect IQ 3.0, которое, как уверяет компания, сильно расширяет функциональность линейки мобильных устройств Garmin. Сервис Connect IQ запущен три года назад; за это время через него выполнено около 54 млн загрузок приложений для более чем 8 млн устройств. В Connect IQ 3.0 для сторонних разработчиков реализована интеграция в совместимые устройства музыки, подкастов, аудиокниги и прочий контент, а в приложения — карт Garmin, с рас-

ширенной системой уведомлений. Также на Developer Summit представлены:

- MySwimPro, предлагающее пловцам индивидуальные 6-, 8- и 12-недельные планы тренировок;
- Yelp by Garmin, помогающее пользователям велосипедных компьютеров Edge найти место для ланча;
- Trailforks, с помощью которого горные байкеры смогут загружать маршруты в свои устройства Edge;
- iHeartRadio, для оффлайн-синхронизации списков воспроизведения пользователями Forerunner 645 Music;
- GU Data Field 2.0 с улучшенным интеллектом логики напоминания, предупреждающее бегуна/байкера, что пора принять очередную порцию энергетического геля.



Garmin Forerunner 645 music, первый пульсометр от Garmin со встроенным музыкальным плеером.

ОБРАЗОВАТЕЛЬНЫЕ ПРОЕКТЫ

Создавая будущее

Два учебных класса в рамках проекта «УчимЗнаем» при поддержке Samsung Electronics открылись в Перми.

Проект «УчимЗнаем», действующий при поддержке компании Samsung, теперь представлен в 9 регионах России — 2 новых класса открылись в Центре онкогематологии при Пермской краевой детской клинической больнице. Цель проекта — создать единое социально-образовательное пространство для детей, находящихся на длительном лечении в стационарах медучреждений. Как технический партнер проекта, Samsung предоставляет необходимое оборудование. Новейшие технологии, а также методики интерактивного и дистанционного обучения, позволят подходить к образованию каждого ребенка индивидуально, с учетом его уровня и состояния здоровья.

Учебные классы «УчимЗнаем» в Перми оборудованы современной компьютерной техникой, в т. ч. сенсорными

телевизионными панелями, компьютерами и планшетами для учеников. В классе проекта «УчимЗнаем» дети смогут полноценно учиться даже на длительном лечении, не только получая новые знания и общаясь со сверстниками на уроках, но и сдавая экзамены и участвуя в олимпиадах. Проект «УчимЗнаем» учрежден в 2014 г. школой №109 г. Москвы («Школа Ямбурга») при поддержке Минобрнауки РФ и региональных ведомств. География проекта быстро расширяется: с 2016 г. участниками «УчимЗнаем» стали детские медучреждения Московской, Ленинградской и Орловской областей, Хабаровского и Пермского краев, Ростова-на-Дону, Калининграда, Красноярска и Воронежа.



На торжественном открытии классов присутствовали министр образования Пермского края Раиса Кассина, заместитель министра здравоохранения Пермского края Константин Шипигузов и руководитель проектного офиса «УчимЗнаем» Сергей Шариков.

АЛЬТЕРНАТИВА ФЛАГМАНУ

Расширение галактик Samsung

Южнокорейская компания официально представила смартфоны Galaxy A6 и A6+.

Samsung продемонстрировала смартфоны Galaxy A6 и A6+ в качестве альтернативы флагманскому Galaxy S9 в среднем ценовом диапазоне. В компании готовы к тому, что продажи S9 и S9+ в будущем останутся «вялыми», а Galaxy A6 и A6+ привлекут покупателей схожей со старшими моделями функциональностью при более низких ценах. Обе модели получили безрамочные дисплеи с закругленными краями (Infinity Display) и соотношением сторон 18:5.9 (такие же установлены на Galaxy S9): 5,6" 1480×720 и 6" 2200×1080 у A6 и A6+ соответственно, а также флагманские задние камеры, 16 МП с апертурой f/1.7 у A6, и дополнительный сенсор 5 МП с апертурой f/1.9 и режимом Samsung Live Focus mode у A6+. Фронтальные камеры у Galaxy A6 и A6+ имеют разрешение 16 МП и 24 МП соответственно. «Под капотом» новых смартфонов находится восьмиядерный процессор с тактовой частотой 1,6 ГГц у A6 и 1,8 ГГц у A6+, батареи емкостью 3000 мА·ч и 3500 мА·ч. Предлагаемые конфигурации — 3 ГБ/32 ГБ и 4 ГБ/64 ГБ ОЗУ/внутренний накопитель. Обе модели оснащаются слотом microSD, работающим с картами емкостью до 256 ГБ

и сканером отпечатков, воспроизводят эффекты Dolby Atmos, имеют встроенный NFC и поддержку Bixby Vision, Home и Reminder (хотя Bixby Voice недоступен). «Мы стремимся предоставить нашим клиентам лучший опыт работы со смартфонами, удовлетворяя все их практические потребности, — заявил Джунхо Парк [Junho Park], вице-президент подразделения Global Product Planning корпорации Samsung. — Galaxy A6 и A6+ используют наследие смартфонов Samsung и сочетают повседневное удобство с впечатляющей камерой, позволяющей владельцам фиксировать, персонализировать и делиться моментами и событиями своей жизни, выражая свою уникальность».

Хотя точная дата поставок еще не объявлена, Samsung заверил, что Galaxy A6 и A6+ будут доступны уже в начале мая, начиная с некоторых европейских, азиатских и латиноамериканских рынков.



Основные козыри нового Samsung Galaxy A6: дисплей а-ля S9 и восьмиядерный процессор.

ИМПОРТОЗАМЕЩЕНИЕ

Сами не плошаем!

Huawei разрабатывает собственные альтернативы Android и Windows 10.

Cтав в 2012 г. объектом расследования структур национальной безопасности США, компания Huawei взялась за свои альтернативы Android и Windows 10. Как сообщает *South China Morning Post*, Huawei «разрабатывает и совершенствует» ОС для смартфонов, готовясь к «наихудшим сценариям». Компания также работает над ОС для планшетов и ПК — на случай, если не сможет ставить Windows на свои устройства. Аналитики считают, что работа над ОС началась в Huawei в 2012 г., когда правительство США назвало Huawei (вместе с другой китайской компанией — ZTE) угрозой национальной безопасности США. В апреле 2018 г. Министерство торговли США на 7 лет, до 13 марта 2025 г., запретило американским компаниям поставлять свою продукцию ZTE, из-за чего ZTE могут лишиться лицензии на использование Android. Планы создания своей ОС инициировал основатель Huawei Рен Чженфэй [Ren Zhengfei].

«Мы разрабатываем собственную ОС по стратегическим соображениям, — цитирует журнал *Caijing*

слова Рена, сказанные им на встрече руководителей Huawei в 2012 г. — Тогда мы не будем обречены, если нам не дадут использовать Android или Windows 8». SCMP сообщает, что Huawei не спешит выпускать свою ОС, поскольку та еще «не так хороша, как Android», к тому же этой пока безымянной платформе не хватает сторонних приложений, разработанных специально для нее. В рамках подобного сценария Huawei разработал собственные процессоры Kirin — на случай, если последует запрет на продажу чипов Qualcomm.

Пока же в заявлении для SCMP представитель Huawei заверил, что «в обозримом будущем выпуск собственной ОС не планируется», добавив: «Мы ориентируемся на продукты под управлением ОС Android, и занимаем открытую позицию по отношению к мобильной ОС». **LXF**

ЭТОЙ ПОКА БЕЗЫМЯННОЙ ПЛАТФОРМЕ НЕ ХВАТАЕТ ПРИЛОЖЕНИЙ СПЕЦИАЛЬНО ДЛЯ НЕЕ.

Сравнение

EasyNAS » FreeNAS » NAS4Free »
OpenMediaVault » Rockstor



Шашанк Шарма

днем — юрист
в суде Нью-Дели,
а ночью судит
открытое ПО!

Дистрибутивы NAS

Ищете возможность создания центрального хранилища-репозитория?
А кто нонче не ищет! **Шашанк Шарма** припас пяток достойных решений NAS.

ПРО НАШ ТЕСТ...

В отличие от настольных дистрибутивов, которым для оптимальной производительности нужны большой объем ОЗУ и хорошая видеокарта, устройство сетевого хранилища (NAS) мало чего требует помимо достаточного места хранения. Готовые решения от производителей обычно предлагают структуру из двух или четырех дисков, и у нас то же самое для наших дистрибутивов, с 4 ГБ ОЗУ.

Все дистрибутивы достаточно просты в установке, однако пост-установочная настройка — подлинное испытание, как и добавление функций через дополнения. Большинство дистрибутивов NAS предлагают простой в навигации сетевой интерфейс администратора, но нам всё же хотелось бы, чтобы он был интуитивным. Система NAS в первую очередь обязана заботиться о данных, и мы обратим внимание на сопутствующие возможности, такие как шифрование данных.

Поскольку суть состоит в разделении дисков между устройствами, надо, чтобы наше решение NAS поддерживало несколько протоколов, типа NFS и AFP. Вследствие изобилия опций настройки важна также понятная документация.



Обычному домашнему пользователю с доступом к одному-двум терабайтам дискового пространства вряд ли нужно сетевое устройство хранения данных (NAS). Хотя использование для хранения данных нескольких дисков — достаточно простое решение для одного компьютера, оно не очень практично для целого дома или офиса с несколькими рабочими местами. А с помощью NAS легко разделить область хранения между пользователями и устройствами.

Довольно часто типовое решение NAS сочетает в себе определенную комбинацию оборудования и программ, специально разработанных для обеспечения общего доступа к файлам через такие сервисы, как NFS и SMB.

Устройства NAS предлагают многие производители; но благодаря дистрибутивам, о которых мы расскажем в нашем *Сравнении*, вы можете настроить свою систему NAS. Однако настройка NAS является куда более сложным процессом, чем простая установка дистрибутива. В отличие от настольных дистрибутивов, они не предлагают вам среды live, чтобы вы могли познакомиться с ними детально, прежде чем решить, какой вам подходит.

Если вы раньше никогда не использовали NAS, выбор дистрибутива может показаться довольно пугающим решением. Будем надеяться, что наше *Сравнение* если не сделает за вас окончательный выбор, то поможет осознать некоторые ключевые факторы перед выбором дистрибутива.

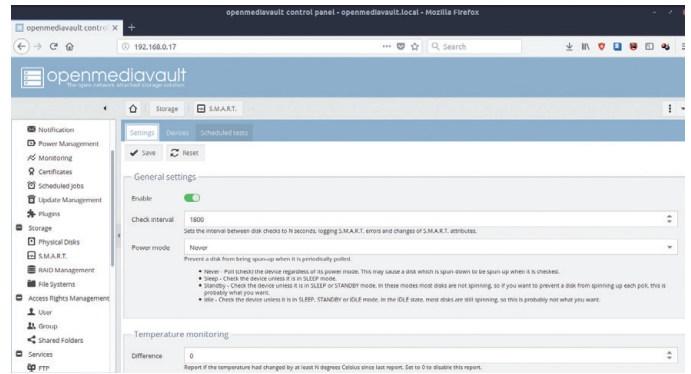
Управление томами

Сработаются ли они с вашими дисками?

Файлы, хранящиеся в центральном хранилище и легко доступные с нескольких компьютеров — главная функция NAS. Поскольку большинство дистрибутивов в нашем списке занимают целый установочный диск, лучше всего, если вы начнете как минимум с двухдискового NAS. Хотя можно разбить ваш диск на разные разделы, установить NAS на один из них и использовать оставшиеся разделы для хранения, это не рекомендуется. Более того, такие дистрибутивы, как FreeNAS и Rockstor, работают только с целыми дисками, а не с разделами.

При выборе места установки вашего дистрибутива NAS помните, что дистрибутивы NAS предлагают web-интерфейс администратора, доступ к которому можно получить с любого компьютера в сети, и таким образом, не предлагают собственной графической среды. А значит, эти дистрибутивы не требуют слишком много места на диске для себя, поэтому вряд ли будет разумно отводить под установку диск с большой емкостью. Почти все дистрибутивы в нашем списке могут практически так же легко быть установлены на USB-диск, что оставляет вам возможность использовать диски только для установки. Вы также можете установить эти дистрибутивы на диск SSD, если у вас есть он под рукой, но они прекрасно будут работать и с обычными дисками. Если раньше NAS вам никогда не требовалось, вы также можете установить эти дистрибутивы в *VirtualBox* и при необходимости добавить виртуальные диски для хранения.

FreeNAS, как многие другие решения NAS, создан на файловой системе ZFS. Она обеспечивает многие из расширенных функций, типичных для NAS, например, целостность данных, создание мгновенных снимков, дедупликация (возможность удалять дублирующиеся копии данных) и т.д.



OpenMediaVault, как и большинство дистрибутивов, предлагает также SMART-мониторинг присоединенных дисков и позволяет планировать тесты.

NAS4Free и Rockstor поддерживают все те же разделы, что и FreeNAS. Благодаря ее функциям копирования при записи и мгновенных снимков, Btrfs считается сильным конкурентом файловой системы ZFS и в EasyNAS и Rockstor является системой по умолчанию.

В дополнение к поддержке файловых систем EXT3/4, XFS и JFS в OpenMediaVault вы также получите опцию создания квот для каждого настроенного тома и списка контроля доступа (ACL).

Выбранная вами файловая система также зависит от функций NAS, которые вы хотите задействовать. Например, многие функции ZFS, такие как дедупликация, очень сильно нагружают ОЗУ.

ВЕРДИКТ

EASYNAS	9/10	OPENMEDIAVAULT	9/10
FREENAS	9/10	ROCKSTOR	9/10
NAS4FREE	9/10		

Все они поддерживают массу файловых систем и сопутствующих функций.

Общий доступ к файлам

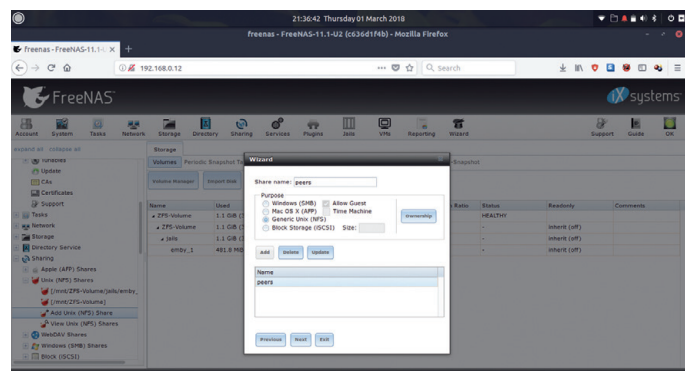
Это — хлеб и масло NAS.

Мало пользы будет от централизованной системы хранения данных, если вы не сможете получить доступ к данным с разных устройств.

Все дистрибутивы поддерживают популярные сетевые протоколы, такие как Samba/CIFS, Apple Filing Protocol (AFP), Network File System (NFS) и Secure File Transport Protocol (SFTP). Хотя по данному вопросу предлагается не слишком много официальной документации, можно также настроить Samba/SSH для обеспечения удаленного доступа к вашим данным через Интернет.

Вы также можете настроить ресурсы общего доступа WebDAV на FreeNAS, позволяя авторизованным пользователям просматривать содержимое определенного тома или директории из удобного браузера. И FreeNAS, и Rockstor поддерживают создание разделяемых ресурсов. Тип создаваемого ресурса будет зависеть от множества факторов, таких как требования безопасности и операционная система, используемая компьютерами в сети. Ради безопасности данных вы должны по каждому ресурсу указать разрешения каждого пользователя. Независимо от выбранного протокола, для введения в разные протоколы, наилучшей работы и полезных советов и рекомендаций мы бы рекомендовали ознакомиться с официальной документацией FreeNAS.

Добавок к названным сервисам, с FreeNAS, OpenMediaVault и Rockstor вы также можете использовать *rsync*. Однако процесс настройки разный



Мастер FreeNAS пригоден для настройки разных аспектов вашей установки NAS, например, разделяемых ресурсов.

для всех дистрибутивов. Например, в Rockstor набором дисков, объединенных вместе и представляющих один том, является Pool. Место, выделенное в Pool, определяется как ресурс, и каждый ресурс в Rockstor ведет себя как директория в настольном дистрибутиве.

ВЕРДИКТ

EASYNAS	8/10	OPENMEDIAVAULT	9/10
FREENAS	9/10	ROCKSTOR	9/10
NAS4FREE	9/10		

Как и FreeNAS, Rockstor также дает возможность определить разрешения доступа для пользователя/ресурса.

Развертывание и управление

Что значит — кривая обучения под углом 90°?!

Xотя NAS традиционно используется для создания центрального репозитория-хранилища, выбранные нами дистрибутивы NAS можно заставить принять на себя куда большую ответственность. Например, вы можете превратить их в систему потокового медиа-вещания или добавить антивирус, чтобы проверять целостность хранимых данных. Поскольку в этих дистрибутивах отсутствует графический рабочий стол, единственным выбором, который у вас имеется — настроить всё вручную из оболочки или использовать веб-интерфейс администратора.

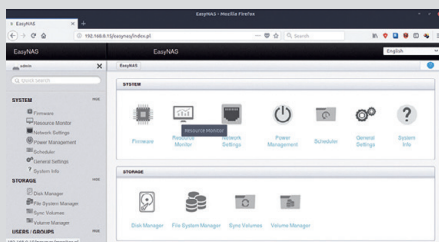
Как бы ни были важны другие функции, нам совершенно ни к чему дистрибутив, с которым невозможно работать из-за неинтуитивности интерфейса. Нам также не нужен и чересчур простой интерфейс — настолько простой, что нельзя отрегулировать настройки.

EasyNAS

4/10

После установки вы попадаете в оболочку, откуда и будете выполнять общие задачи администрирования: изменять пароль администратора или задавать личный IP-адрес. Вы можете получить доступ к веб-интерфейсу, перейдя на <http://<ip-address>:1443>. Игнорируя номер порта, просто направьте браузер на указанный IP-адрес и войдите со своим паролем администратора.

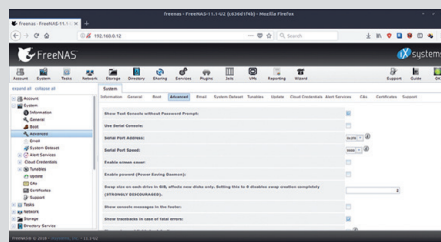
Строку поиска сверху боковой панели можно использовать для поиска строк в Filesystem [Файловой системе], Volumes [Томах], Users [Пользователях] и Groups [Группах]. Т.е. вы не можете использовать ее для того, чтобы вывести настройки NFS, набрав NFS в строке поиска. Боковая панель обеспечивает простой доступ ко всем настраиваемым элементам EasyNAS, поделенным на разные заголовки, например, System Storage [Системное хранилище] и File Sharing [Общий доступ к файлам]. Вы можете просмотреть любую указанную подсказку, нажав на пункт в боковой панели и на кнопку с голубым вопросительным знаком сверху справа.



FreeNAS

9/10

FreeNAS, на базе BSD, во время установки не требует вмешательства пользователя, как и все дистрибутивы в нашем списке. Как и с EasyNAS, через консоль можно настроить сетевые интерфейсы и DNS, а также отрегулировать некоторые другие настройки. Вы можете получить доступ к разным элементам, например, Sharing [Общее пользование] и Plugins [Плагины] по кнопкам в верхней части интерфейса или боковой панели. Там, где это возможно, боковая панель расширяется, отображая подменю, например, Groups [Группы] и Users [Пользователи] в Accounts [Учетные записи]. В качестве функции безопасности дистрибутив дает вам возможность ввести открытый ключ SSH для всех пользователей. Когда вы устанавливаете новый плагин, он автоматически добавляется в боковую панель под заголовком Plugins. Вы можете настроить разные сервисы, нажав на кнопку Services [Сервисы]. Дистрибутив дает возможность запускать сервисы одним нажатием, и даже предоставляет выбор сервисов, которые вы захотите запустить при загрузке.



Документация и поддержка

Когда вы не знаете, в каком направлении движетесь.

Bотличие от настольных дистрибутивов, где можно найти свой путь в системе методом проб и ошибок, NAS не столь благосклонно относится к подобным приемам. Если вы не уверены в том, что делаете, вы легко можете повредить свое NAS, превратив его в бездыханный кирпич.

FreeNAS, один из самых любимых проектов NAS, является сущим информационным монстром. Смело можно сказать, что по части документации FreeNAS для дистрибутивов NAS — то же, что Arch и Gentoo для Linux. Помимо форумов сообщества и канала IRC, на которых можно искать решения определенных проблем, обширная документация размещается на сайте проекта.

Помимо открытых форумов, проект EasyNAS предлагает первичную информацию и краткое введение в важные концепции и технологии, такие как RAID и файловые системы. Однако многие страницы wiki пока что пусты и не содержат контента — например, та, где обсуждается установка EasyNAS на USB.

На сайте NAS4Free вы найдете несколько страниц по таким важным темам, как установка и начальная настройка, и даже по таким экзотическим

темам, как Access Control [Контроль доступа]. Хотя эта информация самая базовая, всё же она полезна.

Как и FreeNAS, OpenMediaVault и Rockstor тоже очень хорошо документированы. Документация OpenMediaVault — только по существу, а Rockstor раскрывает темы подробно и предлагает также множество HowTo по самым разным темам.

В отличие от остальных, FreeNAS предлагает специальную книгу, опубликованную PacktPub. FreeNAS и Rockstor предлагают еще и решения с коммерческой поддержкой, и для каждого из этих дистрибутивов вы также найдете несколько видеоруководств на YouTube.

ВЕРДИКТ

EASYNAS	6/10	OPENMEDIAVAULT	8/10
FREENAS	9/10	ROCKSTOR	9/10
NAS4FREE	8/10		

Rockstor выходит вперед, поскольку предлагает массу скриншотов по всем темам своей документации.

NAS4Free

9/10

OpenMediaVault

8/10

Rockstor

9/10

Консоль NAS4Free позволяет настроить ряд базовых элементов, например, сетевые интерфейсы; но для большинства операций вам придется действовать через web-интерфейс. Однако, в отличие от других web-интерфейсов, NAS4Free не имеет боковой панели. Вместо этого у него сверху размещены вкладки, и наведя на них мышью, вы увидите внутри настраиваемые элементы.

В отличие от дистрибутивов-соперников, NAS4Free предоставляет информацию по многим опциям настройки, например, для настроек CIFS/SMB. Вы можете нажать на Diagnostics > Logs [Диагностика > Журналы], если хотите прочитать отчет о ежедневной работе вашего NAS. В интерфейсе также имеется *QuiXplorer*, полноценный менеджер файлов.

Если у вашего компьютера с NAS есть UPS (ИБП), вы можете настроить его из вкладки Services [Сервисы]. Кроме того, по умолчанию предлагается поддержка LDAP и активной директории.

Основанный на Debian OpenMediaVault предлагает программу установки на базе Curses и требует постоянного участия пользователя, спрашивая о часовом поясе, имени домена, имени хоста, раскладке клавиатуры и т.д. В отличие от других наших дистрибутивов, он не всегда сообщает вам об IP-адресе, с которого можно получить доступ к web-интерфейсу. Кроме того, вы не получаете никаких опций настройки консоли, как в других дистрибутивах.

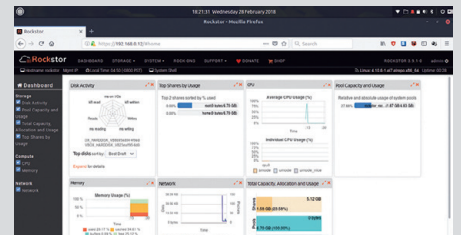
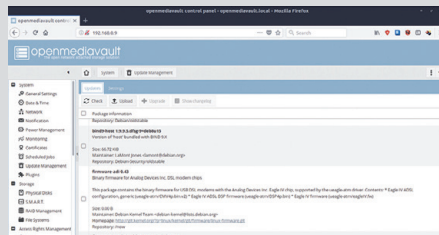
Интерфейс предлагает боковую панель, откуда вы можете получить доступ к разным компонентам, таким, как Access Rights Management [Управление правами доступа], Storage [Хранилище], Diagnostics [Диагностика] и Services [Сервисы]. Вы можете включить и настроить разные сервисы, такие как NFS, FTP и *rsync*, нажав на соответствующий пункт на боковой панели. Однако прежде всего вы должны добавить диски хранилища и смонтировать их. Интерфейс подсказывает вам применить настройки при каждом внесении изменений.

Основанный на CentOS дистрибутив поставляется с установщиком *Anaconda*, и это явный перебор, поскольку всё, что допускается сделать пользователям во время установки — это задать пароль root. Всё остальное, например, использование DHCP для настройки сетевой карты, Rockstor делает автоматически, как и другие дистрибутивы нашего списка.

Web-интерфейс Rockstor похож на NAS4Free и предлагает разные компоненты сверху. Боковая панель активируется, когда вы выбираете компонент, например, Storage [Хранилище].

Как и OpenMediaVault, Rockstor поддерживает массу плагинов, расширяющих функции дистрибутива помимо его скромного предложения NAS. Благодаря использованию Btrfs, он предлагает разные полезные функции, например, резервное копирование и моментальные снимки; но к его использованию Docker для системы плагинов понадобится привыкать.

Кроме всего прочего, дистрибутив потребляет меньше ресурсов, чем FreeNAS.



Функции защиты

Смогут ли они сохранить ваши данные в безопасности?

Преимущество центрального репозитория файлов в том, что вы получаете доступ к вашим данным практически с любого устройства или компьютера в сети. А естественный недостаток — в том, что ваши данные подвергаются риску. Безопасность — ключевая функция любого NAS. Помимо применения шифрования при любой возможности, вы должны настроить надежные правила брандмауэра и списки контроля доступа, чтобы ваши данные не были доступны кому попало. Вы также должны внедрить строжайшие правила безопасности, например, использование HTTPS и безопасных протоколов передачи данных, чтобы ваши данные были в безопасности во время передачи. Неплохой рекомендацией будет также отточить лучшие практики использования SSH, если вы для доступа к своим данным планируете использовать этот протокол.

FreeNAS и NAS4Free основаны на FreeBSD. И хотя сам FreeNAS не обеспечивает особого количества функций безопасности, доверяя пользователям самостоятельно внедрять наилучшие практики, как описано выше, он поддерживает 256-битное шифрование дисков хранилища. Для своих томов ZFS FreeNAS и NAS4Free поддерживают шифрование GELI целого диска. В EasyNAS нет функций безопасности, достойных упоминания. Вы можете

ограничить доступ к интерфейсу администрирования с помощью ACL или IP-адресов и даже настроить Radius, чтобы аутентифицировать удаленных пользователей, и это всё. Неудивительно, что в тесте безопасности этот дистрибутив стоит последним. Как и FreeNAS, Rockstor также поддерживает шифрование целого диска, но с помощью LUKS. OpenMediaVault не дает возможности шифровать данные, но можно установить плагин LUKS для OpenMediaVault, который обеспечит эту функцию. По умолчанию все файловые системы в OpenMediaVault монтируются с флагом `noexec`. Эта функция безопасности не позволяет выполнять скрипты, помещенные в папки с распределенным доступом.

ВЕРДИКТ

EASYNAS	3/10	OPENMEDIAVAULT	8/10
FREENAS	9/10	ROCKSTOR	9/10
NAS4FREE	9/10		

Если бы не его плагины шифрования и безопасности, OpenMediaVault был бы не лучше EasyNAS.

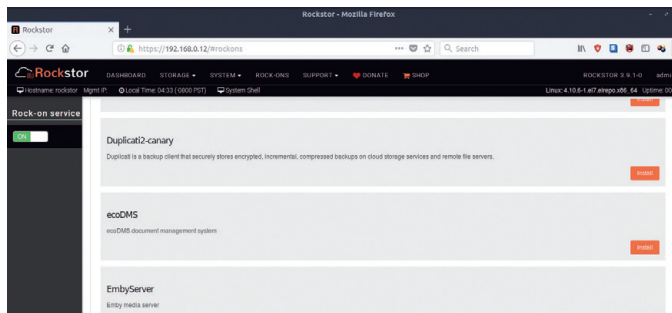
Расширения

Хотите гарнир из плагинов к основному блюду?

Вас расстраивает скудость функций по умолчанию в дистрибутивах из нашего списка? Порадуем вас: об этом волноваться не стоит. За исключением EasyNAS, все дистрибутивы NAS предлагают широкий выбор плагинов. Главное, что для установки этих плагинов вы можете использовать простой web-интерфейс; но будьте бдительны. В отличие от программ установки в настольных дистрибутивах, web-интерфейс не сообщит вам такой информации, как общий объем загрузки для любого плагина. И если вы решите превратить свою установку NAS в сервер потокового вещания медиа, не удивляйтесь, что загрузка сервера *Emby* займет немало времени.

FreeNAS поддерживает ряд сторонних плагинов, таких как Transmission для bittorrent, btsync для синхронизации, с большим количеством устройств без демонстрации ваших данных посторонним, и OwnCloud для размещения собственного сервера облачных сервисов. В порядке меры по укреплению безопасности все плагины во FreeNAS реализованы внутри собственных изоляторов. Список всех имеющихся плагинов вы найдете в официальной документации.

Обратитесь на официальные форумы NAS4Free за информацией и обсуждением функций поддерживаемых плагинов, именуемых расширениями, процесса установки, оптимальных практик и т.п. Одно из самых популярных расширений NAS4Free — OneButtonInstaller (OBI), предназначенное служить централизованным менеджером расширений.



Rockstor предлагает плагины, которые называются Rock-ons через систему Docker и поддерживает обширную коллекцию расширений.

OpenMediaVault предлагает несколько официальных и сторонних плагинов, которые вы можете установить и включить в соответствии со своими потребностями и запросами. Список официально поддерживаемых плагинов помещен под System > Plugins [Система > Плагины]. На странице перечислено более 40 плагинов, разделенных на категории: как Administration [Администрирование], Backup [Резервное копирование], Downloaders [Программы загрузки], Filesystems [Файловые системы], Network [Сеть] и т.д.

ВЕРДИКТ

EASYNAS	2/10	OPENMEDIAVAULT	9/10
FREENAS	9/10	ROCKSTOR	9/10
NAS4FREE	9/10		

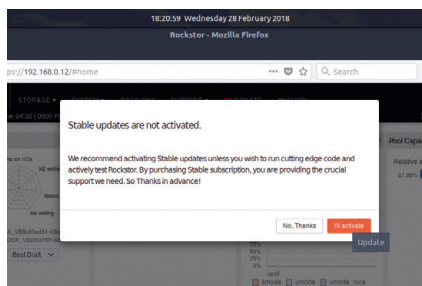
По поддержке плагинов дистрибутивы в этом тесте идут голова к голове. Кроме EasyNAS: его подход – «бери, что дают».

Полезные функции

Что делает их особенными?

Размещение собственного NAS требует большего, чем простая установка основного дистрибутива и отладка пары настроек. Придется принять во внимание и другие факторы, такие как политика обновлений дистрибутива. Это особенно важно, потому что дистрибутивы NAS не предлагают графического рабочего стола.

В EasyNAS легко запланировать автоматическое резервное копирование добавленных томов. К сожалению, здесь нет некоторых расширенных функций — например, нельзя подключиться к серверу директорий. И хотя дистрибутив поддерживает много разных протоколов и сервисов, вы не можете настроить ни один из них. Однако пользователи могут запустить простой web-сервер, что позволяет рассматривать и скачивать файлы из web-браузера. Старейший и любимейший из дистрибутивов NAS, FreeNAS, легко расширяется плагинами. С малыми усилиями вы быстро настроите его для работы с потоковым вещанием медиа или облачными сервисами резервного копирования. Вы можете использовать его встроенный Update Manager [Менеджер обновлений] для применения системных заплаток и новых функций. Одна их уникальных функций FreeNAS — скрипт autotune, автоматически оптимизирующий систему в зависимости от наличного оборудования. Например, если том ZFS существует в системе с ограниченным ОЗУ, скрипт autotune автоматически подстроит некоторые значения ZFS sysctl в попытке минимизировать проблемы с памятью ZFS. Когда скрипт autotune изменяет настройки, измененные показатели отображаются в System > Tunables [Система > Настраиваемое]. Вы также можете запланировать обычные задачи с помощью заданий *cron*. В качестве дополнительной функции безопасности FreeNAS



Интерфейс Rockstor не позволяет игнорировать важные уведомления.

также поддерживает *rsync* через соединение SSH. В OpenMediaVault можно распределять квоты диска отдельным пользователям, ограничивая место, на которое пользователи могут претендовать. Вы также можете определить права доступа каждого пользователя для каждой папки с общим доступом. Как и с FreeNAS, обычные задачи поддержки системы, такие как установка обновлений, можно выполнять из web-интерфейса. OpenMediaVault предоставляет почти все функции FreeNAS и другие, но через плагины. Благодаря файловой системе Btrfs, в Rockstor вы можете создавать Snapshots [Моментальные снимки] и даже это планировать, что достигается с помощью задания *cron*. Это обеспечит безопасность ваших данных, и дистрибутив позволяет идентифицировать те ресурсы общего пользования, моментальные снимки которых вы хотите сделать. Поскольку моментальные снимки инкрементны, у вас должна быть хорошо продуманная политика управления ими, иначе у вас очень быстро закончится свободное место.

ВЕРДИКТ

EASYNAS	6/10	OPENMEDIAVAULT	8/10
FREENAS	9/10	ROCKSTOR	8/10
NAS4FREE	9/10		

Как и в других тестах, набор функций NAS4Free такой же, как у FreeNAS.

Дистрибутивы NAS Вердикт

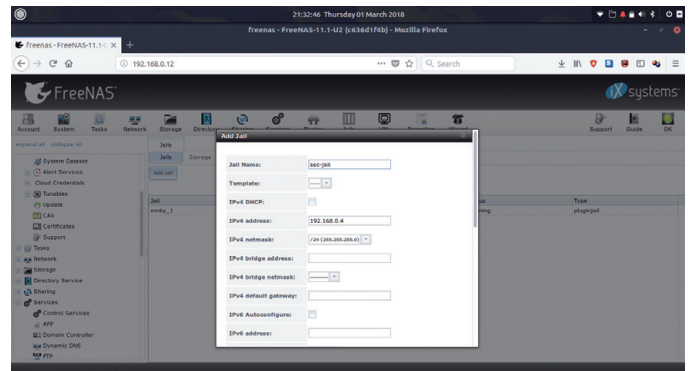
Среди пользователей бытует мнение, что на нормальную настройку NAS, в зависимости от ваших требований, может уйти неделя — ну уж точно не один день. Выбранные нами дистрибутивы предлагают все функции и гибкость коммерческих NAS, но без расходов на проприетарное ПО.

EasyNAS, несмотря на свой простой в использовании интерфейс, оставляет желать лучшего. Нажав на NFS на боковой панели, вы можете включить/отключить сервис в главной панели и создать новые ресурсы NFS. К сожалению, нельзя редактировать настройки NFS, несмотря на наличие вкладки Settings [Настройки]. При нажатии на нее появляется пустое окно, и невозможно открыть файл настроек или написать собственный. Строка поиска тоже ненадежна и не показывает результатов, даже если ввести точное имя. Кроме того, дистрибутив предлагает мало опций при создании томов, и у него даже нет шифрования или поддержки плагинов. Именно по этим причинам EasyNAS, несмотря на свою простоту в работе, приходит последним.

Хотя NAS4Free предлагает куда больше настраиваемых элементов, чем другие дистрибутивы по умолчанию, его интерфейс прост и логичен, и не кажется загроможденным. Хотя дистрибутив «просто работает» и делает всё правильно, занять место на пьедестале ему не позволила ограниченность его документации.

К сожалению, ряд аспектов интерфейса OpenMediaVault, таких как определение квот пользователей, несколько контр-интуитивен. Квоты определяются не из окна управления пользователями, а из управления дисками, отдельно для всех доступных дисков. Правда, это скорее непривычно, чем неудобно. В отличие от других дистрибутивов, у OpenMediaVault есть функция блокировки по умолчанию; это удобно, если вы, зарегистрировавшись в веб-интерфейсе, куда-либо отходите. Однако из-за чрезмерного упора на плагины для обеспечения основных функций этот блестящий во всех остальных смыслах дистрибутив занимает только третье место.

Использование во FreeNAS механизма FreeBSD, Jails [Изолятор], для запуска плагинов внутри изолированных хранилищ гарантирует, что даже если плагины будут взломаны, они не повлияют на NAS. Но если вам приходится выбирать NAS для компьютера с ограниченным ОЗУ, лучше выбрать Rockstor. **LXF**



1^е место FreeNAS **9/10**

Версия: 11.1 **Сайт: www.freenas.org** **Лицензия: BSD**
Сильно нагружает ресурсы, но зато предлагает много функций.

2^е место Rockstor **8/10**

Версия: 3.9.1 **Сайт: Rockstor.com** **Лицензия: GPL**
Кабы не реализация во FreeNAS механизма Jail, быть бы ему победителем.

3^е место OpenMediaVault **6/10**

Версия: 3.0.94 **Сайт: www.openmediavault.org** **Лицензия: GPL v3**
Слишком сильно опирается на плагины, обеспечивая стандартные функции.

4^е место NAS4Free **6/10**

Версия: 11.1 **Сайт: www.nas4free.org** **Лицензия: BSD**
Имеет больше общего с FreeNAS, чем его основа FreeBSD.

5^е место EasyNAS **4/10**

Версия: 0.7.2 **Сайт: www.easynas.org** **Лицензия: Разные открытые**
Хорош для обычных домашних пользователей, но продвинутых функций нет.

» РАССМОТРИТЕ ТАКЖЕ

В нашем *Сравнении* мы рассказали о лучших решениях NAS. Имеется длинный список пассивных решений NAS, которые мы не можем рекомендовать, потому что они уже довольно давно не обновлялись. Помимо решений с открытым кодом, есть и коммерческие решения. Большинство берут оплату за функции уровня предприятия, а некоторые — за поддержку и прочие удобства.

ipRAID Server доступен в виде образа USB. Вы можете попробовать его бесплатно в течение 30 дней, после чего вам придется купить

лицензию от \$59 (примерно £43), позволяющую подключение до шести устройств. Open-E DSS V7, со множеством корпоративных функций, также предлагает 60-дневный пробный период.

Далее имеется Server Elements с тремя продуктами NAS, работающими полностью из ОЗУ. NASLite-2 является общецелевым решением NAS с розничной ценой \$30, тогда как NASLite-M2 за \$35 специализируется в потоковом вещании медиа, а NanoNAS за \$10 разработан для домашних пользователей.

RASPBERRY Pi И РОБОТЫ!

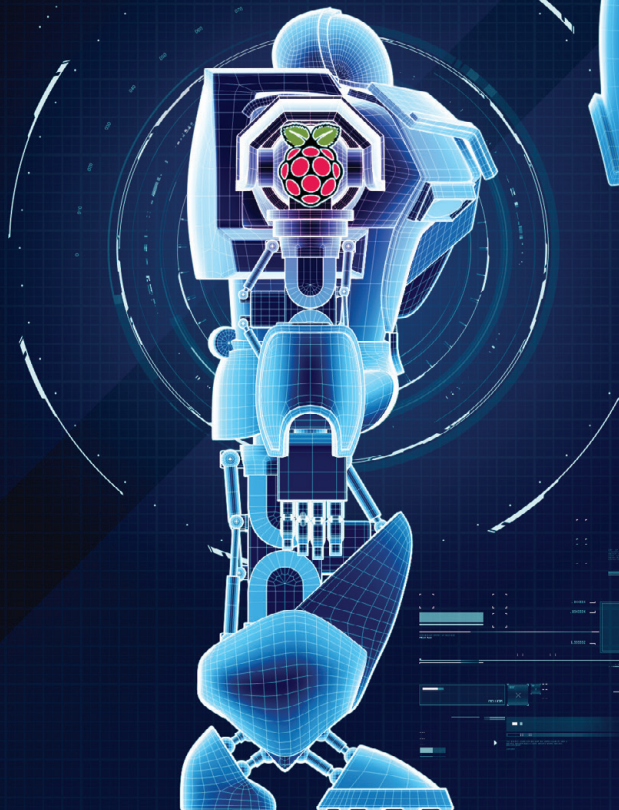
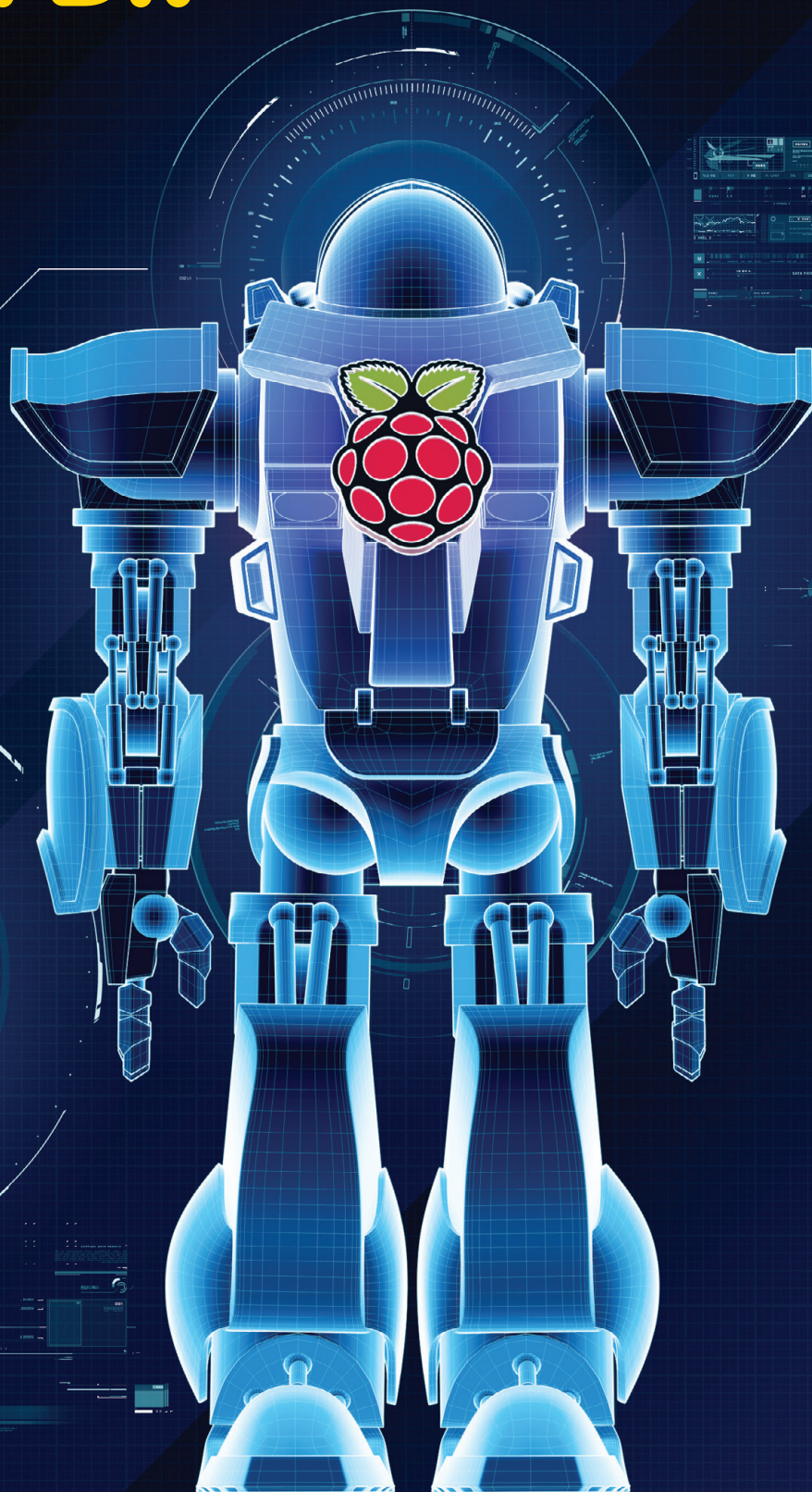
Суперзлодей **Джонни Бидвелл** создает армию роботов на Pi. Пробреритесь в его логово и украдите его технологии!

Робототехника не обязательно подразумевает создание жутких человекоподобных автоматов. Мы можем создавать устройства, способные выполнять самые разные полезные или интересные функции. И, что еще лучше, их можно использовать с Linux.

Для создания этих устройств не нужна дорогостоящая электроника. Прекрасно подойдет Raspberry Pi. И это не должно быть сложнее сборки набора LEGO. Есть множество готовых комплектов, которые устроят любого энтузиаста.

Мы расскажем, как собрать собственного ходячего робота, и покажем вам пару симпатичных программ Python для начала работы. А если вы наберетесь смелости, то почему бы самостоятельно не построить собственных роботизированных помощников? Все компоненты легко найти благодаря растущему сообществу умельцев.

Если вы намерены удариться в крайности, тогда напечатайте на 3D-принтере необходимые части и с помощью компьютерных заклинаний добавьте своему созданию псевдо-эмоции. Или создайте собственного домашнего любимца-робота, и пусть он повсюду трусит за вами, как щенок. Ваши возможности безграничны. Так что включайте Pi, находите компоненты и давайте вместе примемся за изготовление роботов.



Потенциал Pi-бота

Мы говорим не о создании T-101: Pi Edition а-ля Шварценеггер, однако пусть вас это не расхолаживает.

Raspberry Pi появился шесть лет назад, и с тех пор превратился в нечто удивительное. Мы видели четыре основных версии (Zero, 1, 2 и 3), и парочку младших версий; самая последняя — мощный 3 В+, выпущенный в марте 2018 г.

Продавец Pi, ModMyPi.com, действовал с самого начала, и любезно предоставил нам несколько деликатесов для этой статьи. Его основатель Джейкоб Марш [Jacob Marsh] делится своим восприятием удивительного успеха Pi: «Я не думал, что Raspberry Pi превратится в то, чем является сейчас, я даже не думал, что это произойдет с Pi Foundation. ModMyPi был зачат в спальне университетского кампуса. А теперь у нас всемирный электронный магазин с более чем 2000 линиями продуктов, оборотом в £3 миллиона, семью штатными сотрудниками и складом площадью 300 квадратных метров, которого нам очень быстро стало мало. Да продолжится это в том же духе и в будущем!»

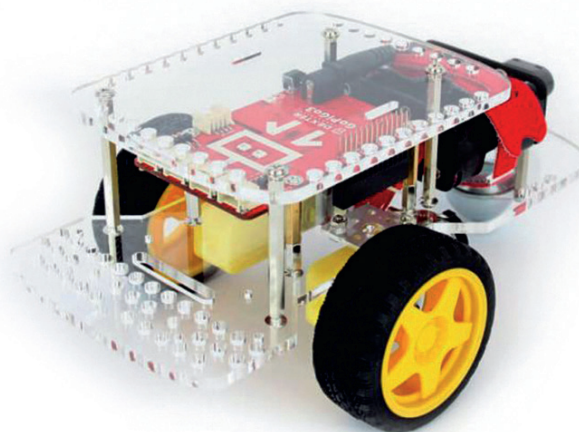
Что превращает Pi в нечто уникальное — и что способствовало нехватке площадей недвижимости у Джейкоба — это безграничные возможности расширения. Крошечный компьютер общецелевого назначения, который способен подключаться к датчикам, моторам или системе центрального отопления, вкупе с коллективным воображением участников сообщества открытого кода привели к появлению разнообразных физических компьютерных устройств. И одно из самых удивительных таких устройств — роботы.

» РОБОПЛАТФОРМЫ

Мы сосредоточимся на самых основных программах, а именно, Raspbian и Python, однако есть ряд интересных платформ, которые можно запустить поверх, вместе или вместо них.

Одна из таких платформ — Robot OS (ROS, см. www.ros.org), которая на самом деле вовсе не операционная система, а скорее промежуточное ПО. ROS — проект под лицензией BSD, с сотнями участников по всему миру. Его можно скомпилировать с нуля для Raspbian, но поскольку он предлагает пакеты для редакции Pi Ubuntu Mate, эта основа лучше. ROS предназначен стать истинно общецелевым ПО для роботов, которое работает на множестве платформ и всячески способствует сотрудничеству.

Нас также впечатлила работа Dexter Industries (www.dexterindustries.com), которые выпускают машину-робот GoPiGo. GoPiGo — это комплект для школ, который программируется на Bloxter, основанном на браузерном языке, похожем на Scratch. Внутри интерфейса Blockly более опытные ученики могут также программировать на Python. Фанатам LEGO понравится Dexter BrickPi, подключаемый к Raspberry Pi с помощью популярного набора LEGO Mindstorms. Мало того, он еще составляет Grove Pi, набор из 12 подключаемых сенсоров для исследования программ IoT.



Робот Dexter GoPiGo — популярный выбор для образования, и он явно не помешает о мировом господстве.

Не поддавайтесь намеку забавной иллюстрации, открывающей статью. Любая ситуация, где управляемое компьютером устройство выполняет механические функции, по сути своей является роботом. На данный момент среди любителей Pi особой популярностью пользуются машины-роботы. На первый взгляд они мало чем отличаются от радиоуправляемых машинок, однако способны не только развезать вокруг кухонного стола.

Наличие внутри крошечного устройства Linux открывает широчайшие возможности. С рядом легко подключаемых компонентов, библиотеками с открытым кодом и небольшим программированием на Python, чтобы связать всё это воедино, вас ограничивает только ваше воображение. Да, небольшая поправка: возможно, вам придется припаять несколько проводов. Поначалу это пугает, но в Интернете полно руководств. Посмотрите, как

ДЕРЖИТЕ РОБОТА ПРИ ДЕЛЕ

«Работающий на Pi робот может также делать фотографии, бродить по дому, цитируя Шекспира, или даже взмыть в небеса.»

директор по образованию Pi Foundation, Кэрри-Энн Филбин [Carrie Anne Philbin] устраивает отличную демонстрацию на <https://youtu.be/P5L4G16Q4Xo>.

Робот Pi может работать на том же Raspbian Linux, который мы все знаем и любим. Он способен подключаться к беспроводным сетям, так что вы можете входить через SSH и проводить диагностику на ходу. Вы даже можете запустить на ходу `apt upgrade`. Работающий на Pi робот способен также делать фотографии, бродить по дому, цитируя Шекспира, или даже взмыть в небеса, чтобы восхититься открывающимся внизу видом города. С помощью библиотеки OpenCV и небольшой работы с изображениями мы можем дать нашему роботу компьютерное зрение, чтобы он мог определять и следовать за объектами или даже распознавать объекты и людей. Итак, изучайте наше руководство и не забудьте заглянуть на www.modmypi.com в поисках вдохновения.

Создание DiddyBorg

Создавать роботов проще, чем вы думаете. Давайте мы покажем вам, как...



Добрые люди из ModMyPi были так любезны, что прислали нам DiddyBorg v2 поиграть. Мы безгранично им за это благодарны и настоятельно рекомендуем вам заглянуть в их магазин, когда вам в следующий раз понадобится что-нибудь, связанное с Pi.

Серия Borg была разработана в PiBorg, детище Тимоти Фриберна [Timothy Freeburn], который объясняет: «DiddyBorg — шестиколесный робот, дань уважения марсоходу Sojourner». PiBorg начал сотрудничать с ModMyPi в октябре 2017 г. Джейкоб так объясняет это сотрудничество: «Мы с Тимом оба были свидетелями зарождения первого Raspberry Pi В. Я занимался созданием корпусов, а Тим создавал регуляторы двигателя и всякие LED (www.piborg.org/ledborg/), и когда ModMyPi дорос до солидной розничной торговли, продукты Тима были одними из первых, которые мы выбрали.

По мере роста рынка мы осознали необходимость больше развивать внутреннюю разработку, а Тим осознал ценность нашей сети распространения для своей продукции. Он смог уделить больше времени своим крутым продуктам, а мы смогли уделить больше времени продаже и рассылке. Мы оба хорошо работаем над достижением наших целей и хорошо ладим, так что начать сотрудничество было несложно!»

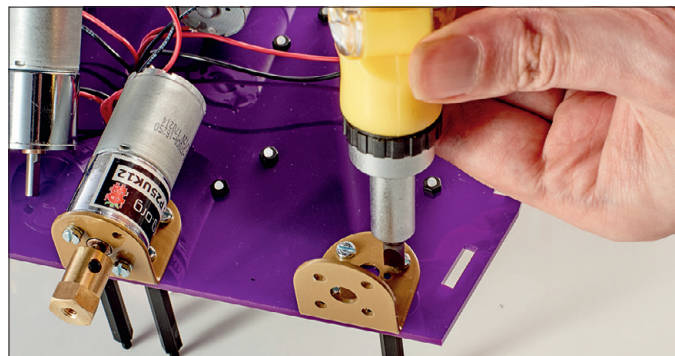
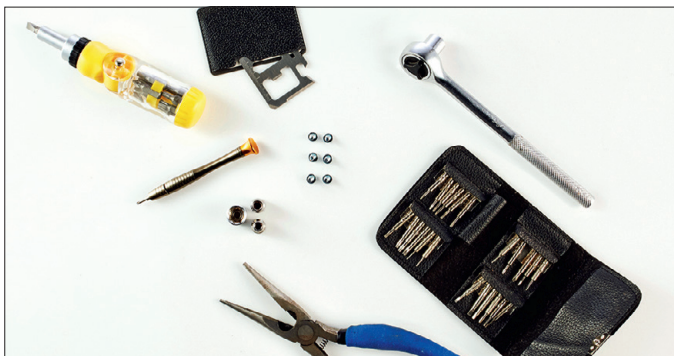
В нашем DiddyBorg шесть мощных моторов по 12 В на 100 об./мин., и он работает на 10 батарейках AA. Далее мы рассмотрим альтернативные источники питания. Питание подается со специальной ThunderBorg, которая может выделить 5 А на каждый мотор

и снабжена цветными ЖК-диодами [LED] (что удобно для мониторинга состояния батареек), и мы уверены, что вы согласитесь: выглядит это вполне как бизнес-класс. Вместо использования для управления сложного сервопривода, DiddyBorg поворачивает посредством вращения колес с разной скоростью.

Борги (см. ниже) и многие другие роботы продаются как наборы для самостоятельной сборки. В зависимости от комплектации это может быть довольно сложный процесс, включающий мелкие части, пайку и даже, возможно, нецензурные выражения. На создание Diddy от начала до конца уходит пара часов, и вы найдете инструкции на www.piborg.org/blog/build/diddyborg-v2-build/diddyborg-v2-buildinstructions. Было бы глупо полностью воспроизводить их здесь, однако обратите внимание на наше общее пошаговое руководство по сборке робота на стр. 30. Мы создали его на основе DiddyBorg, потому что у нас он есть; но создание других роботов идет по такому же рецепту.

Если вы такое умеете, неплохо будет перед началом сборки оптимальным способом настроить Pi. Конечно, можно войти в робот по SSH и после сборки, чтобы всё окончательно настроить (поскольку вы заранее сообщили ему о вашей беспроводной сети), однако интенсивное использование беспроводного соединения будет потреблять мощность батарей, которую можно использовать, чтобы промчаться по офису, раздражая команды из других журналов. Если у вас есть еще один Raspberry Pi и ловкость рук, можете после сборки извлечь SD-карту и выполнить всю настройку на другом устройстве.

СОЗДАНИЕ РОБОТА Pi

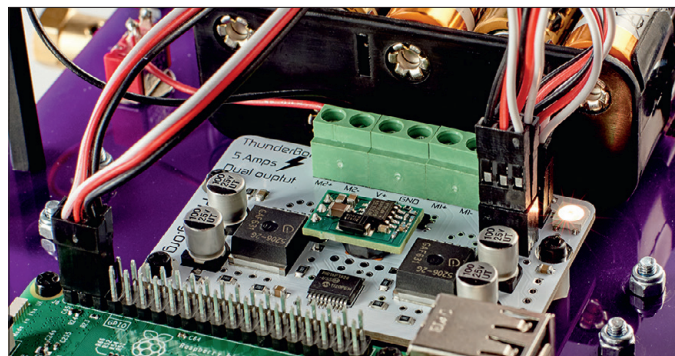


1 Организуйте свое рабочее место

Мы знаем, как соблазнительно сразу приняться за дело и вывалить содержимое всех упаковок на ваш и без того захламленный стол. Не надо! Приберите на своем столе, освободив рабочее место, внимательно прочтите инструкции и разложите все необходимые инструменты с одной стороны. Если придется что-то паять, убедитесь, что помещение хорошо проветривается и у вас под рукой тампон для удаления припоя.

2 Крепеж

Обычно моторы прикрепляются к нижней стороне шасси несколькими угловыми скобами. Найдите нужные отверстия и закрепите эти скобы соответствующими болтами и гайками. DiddyBorg использует металлические зажимы для закрепления этих скоб. Не забудьте добавить шайбы, если они нужны на шасси. Они помогут распределить усилие и защитят шасси от поломок.

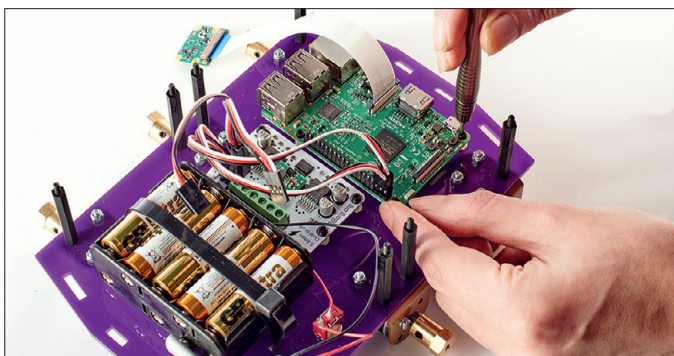


6 Присоедините моторы и контроллеры

Закрепите моторы на месте с помощью винтов. DiddyBorg использует пружинные шайбы для обеспечения сцепления, даже если ослабнет держащий винт: ведь вам совершенно ни к чему, чтобы у робота по дороге отвалился мотор. После этого присоедините контроллер мотора к шасси или сначала прикрепите требуемые стойки. Примите во внимание ориентацию контроллера.

4 Паяйте и присоединяйте

Нарежьте провода нужной длины и зачистите концы. В этой небольшой врезке нам не удастся рассказать вам, как паять, так что будьте осторожны. Припаяйте красные провода (если надо) к положительным сторонам моторов, а черные — к отрицательным. Используйте фольгу, чтобы защитить шасси от капель припоя. Подсоедините провода к соответствующим клеммам контроллера.



5 Подключите свой Pi и батарейки

Прикрепите отсек для батарей и выключатель к шасси и затем подключите их к контроллеру. У роботов помощнее это может включать более сложную работу с проводами по сравнению с моторами. Проверьте наличие SD-карты в Pi (и что Raspbian настроен на использование вашего беспроводного соединения и контроллера Bluetooth) и подключите Pi к шасси. Не слишком сильно затягивайте винты.

6 Завершение работы

Подключите контроллер к требуемым контактам GPIO на Pi (проверьте, правильно ли ориентированы коннекторы). Подключите остальные периферийные устройства робота (например, сенсоры и камеры). Надежно прикрепите втулки колес к моторам с помощью установочных винтов и проверьте наличие достаточного зазора для свободного вращения. Наденьте колеса на втулки и надежно закрепите их на месте.



Найдите роботу занятие

То, что можно сделать с роботом, ограничено только вашим воображением и временем. И вот вам несколько предложений для вдохновения...

Собрав нашего Diddy, мы хотели больше узнать о том, что он может делать. Не удовлетворившись созданием прекрасного оборудования, PiBorg предлагает также отличные примеры кода, чтобы вы могли приступить к работе. Для работы на другом оборудовании потребуется небольшая настройка, однако это должно дать вам некоторое представление о том, как общаться с оборудованием на Python.

В комплекте с роботом предоставлены скрипты для обработки всей низкоуровневой коммуникации. В нашем случае это делается через файл **ThunderBorg.py** (см. www.piborg.org/blog/build/thunderborg-build/thunderborg-examples). Он работает со всем кодом I2C, так что вам незачем об этом беспокоиться, и предлагает

еще больше удобных для человека функций — например, функцию `SetMotor1()`, которая задает скорость колес с левой стороны.

Web-контроль

Предположим, ваш Pi подключен к беспроводной сети; тогда один из немногих не прямых способов управлять им — запускать его через небольшой web-сервер с формой HTML для контроля. Если к вашему роботу подключена также и камера, вы можете передавать поток видео на ту же страницу, чтобы насладиться чудным восприятием вождения от первого лица.

Создание потокового видео процессора в Python менее сложно, чем можно подумать, но всё же сложнее, чем нам бы хотелось для данного обзора. Поэтому изучите пример сетевого пользовательского интерфейса DiddyBorg на www.piborg.org/blog/build/diddyborg-v2-build/diddyborg-v2-examples-web-ui, чтобы увидеть, как творится волшебство. Если вам повезло обзавестись DiddyBorg, скопируйте туда этот скрипт.

Управление геймпадом

Освоить управление вашим роботом с геймпада немного проще. Однако, как мы обнаружили, могут потребоваться некоторые труды, чтобы заставить всё работать, если не запущен GUI (например, если ваш Pi не подключен к монитору). В теории вы могли всё настроить заранее, или вытащить SD-карту из робота и загрузить ее на другом Pi — настройки должны запомниться. Если нет, мы можем всё настроить, войдя в робот через SSH.

Нужно справиться с двумя задачами: подключением Bluetooth и последующей настройкой узлов устройства. Последнее делается с помощью пакета джойстика — *joystick* (или *evdev*, от которого он зависит), а первое — с помощью команды `bluetoothctl` (стандартно она будет установлена). После установки пакета *joystick* запустите `bluetoothctl`. Запустится консоль, где можно сканировать, коммутировать и подключать наш контроллер.

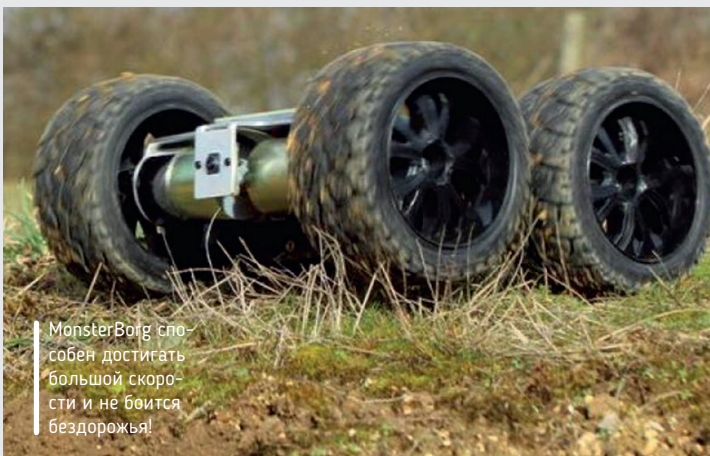
Сначала переведите устройство в режим коммутирования, а затем начните сканирование — командой `scan on`. Вы должны увидеть список всех расположенных поблизости устройств Bluetooth и их MAC-адресов. Будем надеяться, ваш контроллер на месте, и в таком случае скопируйте адрес. Деактивируйте

»» СЕРИЯ BORG

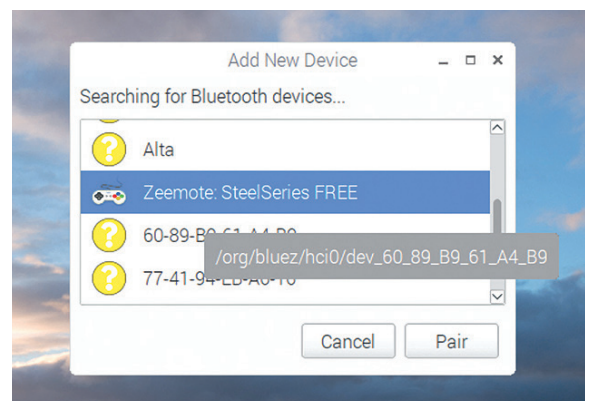
Сопротивление бесполезно. Серия PiBorg Borg состоит из трех роботов: DiddyBorg (£210), MonsterBorg (£210) и YetiBorg (£160). Совершенно плоский верх DiddyBorg дает возможность смонтировать на нем всё, что угодно. Например, жуткий механический коготь. Его можно перенастроить с помощью Multifunction Top, чтобы разместить дополнительный Pi, или с помощью Touch Top, задействовав официальный сенсорный экран Raspberry Pi. На нем также можно разместить камеры переднего и заднего вида Pi Cameras и ультразвуковые сенсоры, чтобы избегать препятствий.

Далее идет MonsterBorg, устройство, с которым не стоит шутить. Массивные четырехдюймовые колеса позволяют ему вгрызаться в бездорожье. Возможно, вы читали в разделе Новостей, что PiBorg организует гонки Formula Pi, а именно в гонках MonsterBorg просто лучший. Его 3-миллиметровые алюминиевые шасси позволяют ему пережить всё, кроме самых убийственных столкновений.

И, наконец, скромный YetiBorg. Работающий на Pi Zero (включен) и одной батарее 9В, YetiBorg использует индивидуальный контроллер мотора ZeroBorg, позволяющий индивидуально управлять его четырьмя моторами. Его можно собрать в двух разных конфигурациях: низкопрофильной и внедорожной. В низкопрофильном режиме, если он случайно перевернется, он прекрасно сможет продолжать работу вверх ногами. Если требуется чуть больший зазор, то внедорожный режим защитит шасси от неровностей, а электронику от брызг.



MonsterBorg способен достигать большой скорости и не боится бездорожья!



Геймпады Bluetooth легко настроить с рабочего стола Pixel. Вот бы всё было так же легко в командной строке...

сканирование по scan off. Затем свяжите с помощью pair <MAC-адрес>, подключите с помощью connect <MAC-адрес> и выйдите с помощью exit.

Теперь запустите evtest, который поприветствует вас списком обнаруженных устройств ввода. Выберите нужный контроллер и жмите на кнопки. Вы увидите разный криптографический ответ для каждой кнопки. DiddyBorg включает примерный скрипт для управления джойпадом, который использует библиотеки PyGame, чтобы слушать соответствующие события кнопок.

Распознавание изображений

Наш второй материал в этом месяце полностью посвящен машинному обучению, и если вы его читали, вы заметили, что мы упомянули запуск TensorFlow на Pi. Всё это благодаря работе Сэма Эйбрахамса [Sam Abrahams], который предоставил прекомпилированные файлы колес для Python 2.7 и 3.4. Это хорошая новость, если у вас предпоследняя версия (Jessie) Raspbian, поскольку она включает Python 3.4. Однако если вы установили себе самую последнюю версию (Stretch, где используется Python 3.5), то вам надо взять Python 2.7.

Наличие двух разных версий Python, как здесь, вполне нормально (Raspbian стандартно предлагает обе), но нельзя установить одновременно 3.4 и 3.6, и колесо 3.4 не будет работать с Python 3.6. Прежде чем приступить к работе, запомните, что репозиторий моделей Tensorflow объемист, и при использовании SD-карты на 8 Гб у нас не раз заканчивалось место. Это можно обойти, удалив большие пакеты, например, LibreOffice и Wolfram Alpha, но советуем использовать 16-Гб карту. Следующие команды настроат всё, что вам нужно:

```
$ wget https://github.com/samjabrahams/tensorflow-on-raspberry-pi/releases/download/v1.1.0/tensorflow-1.1.0-cp27-none-linux_armv7l.whl
$ sudo apt install python-pip python-dev python-pil python-matplotlib python-lxml
$ sudo pip install tensorflow-1.1.0-cp27-none-linux_armv7l.whl
$ git clone https://github.com/tensorflow/models.git
```

На последнем этапе запустится загрузка объемом этак 1 Гб, так что будьте внимательны. Если место у вас закончится, процесс можно возобновить, как только вы расчистите место, запустив git checkout -f HEAD из директории models/. После его успешного завершения проверьте его с помощью

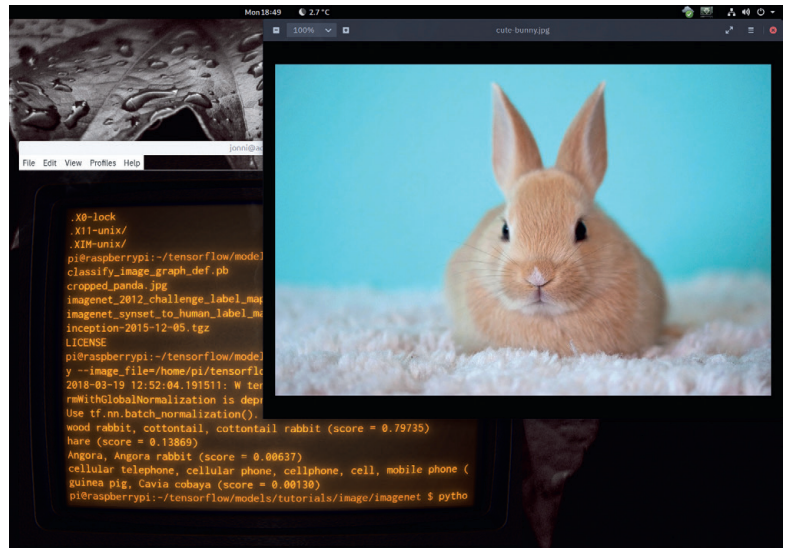
```
$ cd models/tutorials/image/imagenet
$ python2 classify_image.py
```

У вас должно определиться изображение панды (оно было распаковано в /tmp/imagenet/cropped_panda.jpg). Скрипт также может принять параметр --image-file для определения представленных пользователем изображений. Так что, немного повозившись, мы сможем всё адаптировать, чтобы сделать фотографию и потом попытаться ее определить. Поскольку на Pi 3 весь процесс занимает около 10 секунд (хотя его можно ускорить, прогоняя программу в цикле, используя C++ или с помощью более скромной модели TensorFlow), мы не питаем особых надежд на классификацию в реальном времени.

Более того, вероятно, что скрипту будет довольно трудно определить фотографию, сделанную на уровне пола с помощью Pi Cam. Но это не страшно и только подбавляет интереса. Всё, что надо сделать — немного поднастроить classify_image.py. Скопируйте этот файл в classify_photo.py, затем отредактируйте новый файл. Вам надо на ранней стадии импортировать модуль picamera, затем в функции main() замените строку, которая начинается с image =, на нечто подобное:

```
cam = picamera.PiCamera()
cam.capture('/tmp/picam.jpg')
```

И, наконец, измените вызов run_inference_on_image() для работы на нашем недавно сделанном picam.jpg. Если вам хватит



TensorFlow распознал здесь кролика, но потом Джонни счел его картофелиной, Нейл — галстуком-бабочкой, а Эффи решил, что это боксерская груша.

смелости, то почему бы не привязать новый скрипт к событию нажатия на кнопку? Скрипт контроллера для DiddyBorg, который мы обсуждали ранее, легко может адаптироваться для подобной цели.

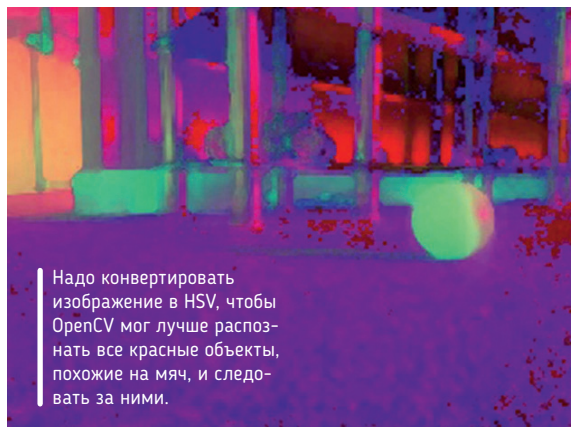
Следим за мячом

OpenCV — мощная среда компьютерного зрения, которая включает привязки Python. Она использовалась для создания изображения в ранее приведенном примере Web UI, но мы можем использовать ее и для более сложных целей. Она может опре-

СЛЕДИТЕ ЗА РОБОТОМ

«Если к вашему роботу подключена также и камера, вы можете наслаждаться чудным восприятием вождения от первого лица»

делять объекты на изображении, что означает возможность направлять к ним нашего робота. Если эти объекты движутся, робот следует за ними. Чтобы это сработало, указанные объекты должны быть хорошо различимы, как, например, яркий мяч. Именно такой пример вы найдете на www.piborg.org/blog/diddyborg-v2-examples-ball-following.



Надо конвертировать изображение в HSV, чтобы OpenCV мог лучше распознать все красные объекты, похожие на мяч, и следовать за ними.



Экипируйте своего робота

Робота можно оснастить множеством разных странных и чудных виджетов, чтобы добавить ему функциональности, да и просто удовольствия ради!

Простому роботу Pi мало что нужно, помимо шасси, источника питания и колес. Однако всё становится куда интереснее, если оснастить базовую конструкцию дополнительными периферийными устройствами. Можно новыми, новаторскими способами использовать хорошо известные периферийные устройства Pi, такие как камера и сенсорный экран, или использовать другие приспособления — сенсоры, микрофоны, сервомоторы и т.п. — чтобы достичь потрясающего уровня.

Найдите контроллер

Управлять роботом через web-интерфейс хорошо и мило, хотя даже в самой быстрой сети у вас будет некая степень задержки, что сделает движение немного беспорядочным. Проводной контроллер будет работать надежно, но понятно, что от него не слишком много пользы, если вы не сможете следовать сразу за своим роботом. С помощью контроллера Bluetooth вы можете добиться

ВСТАВЬТЕ РОБОТУ ГЛАЗА «Если камере нужна гибкость, то доступны разные наборы поворотников, позволяющие добиться идеального угла.»

куда более точного контроля над своим другом-роботом. Джойпады PlayStation 3 очень популярны при использовании вместе с утилитой *sixpair*. Более подробную информацию вы найдете на www.piborg.org/blog/rpips3-help. Однако есть ряд других опций, и с Raspbian многие будут работать по умолчанию. Устройства Bluetooth класса 1 в теории имеют радиус действия до 100 м, но они подвержены помехам и перебоям в связи.

Также, если у вас нет Pi 3 или адаптера Bluetooth, это в любом случае не вариант, так что стоит рассмотреть возможности использования альтернативных средств приема и передачи.



Мы обнаружили этот удобный маленький геймпад Bluetooth в одном из пыльных углов редакции Future Towers. Он здорово пригодился для данного материала.

Хорошей опцией здесь будет использовать радиочастоту (RF) в полосе 2,4 ГГц. Это потребует более мощного передатчика, например, такого, который используется в дронах и других радиоуправляемых летательных аппаратах.

Используйте Pi Camera

Чего бы вы ни хотели от своего робота, вряд ли ему помешает возможность видеть. Это легко обеспечить с помощью платы Pi Camera, доступной в магазинах для умельцев. Запущенный в 2013 г., Pi Camera Module был одним из первых официальных дополнений для Pi, и стал использоваться в самых разных интересных проектах. В 2016 г. вышла новая восьми-мегапиксельная версия, а за ней последовала мини-редакция для Pi Zero.

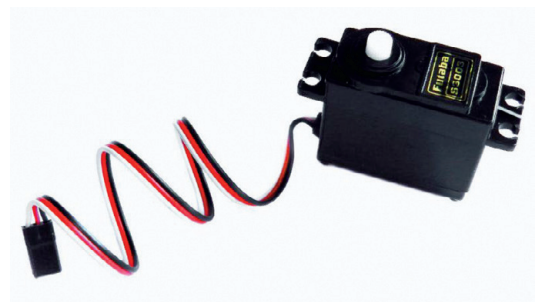
Эти камеры подключаются через шлейф к интерфейсу CSI, оставляя контакты GPIO свободными для чего-нибудь другого. Крепеж доступен так, чтобы можно было надежно установить камеру там, где требуется, и если камере нужна гибкость, то доступны разные наборы поворотников, позволяющие добиться идеального угла. Кроме того, имеются телефото, fish-eye и широкоугольный объективы. На предыдущих страницах мы видели, как можно использовать библиотеки вроде OpenCV для определения объектов с помощью входящей информации с камеры. Если вы хотите использовать Pi Camera в своих проектах с камерой, вам поможет такая команда в одну строку:

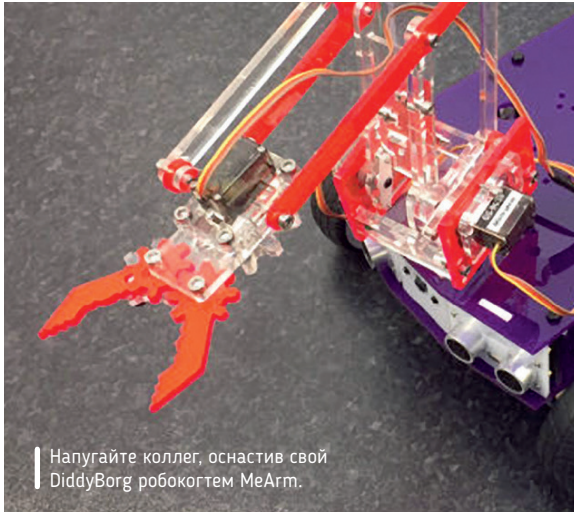
```
$ sudo apt install python-picamera
```

Добавьте движения сервомоторами

Сервомоторы — это моторы [Ред.: — На самом деле, исполнительные механизмы], которые часто имеют ограниченный диапазон движения, например, чтобы перемещать руль радиоуправляемой лодки, но некоторые умеют — или их можно под такое адаптировать — выполнять непрерывное вращение.

Сервомоторы управляются широтно-импульсной модуляцией (PWM). Эти импульсы должны быть довольно точными, иначе сервомоторы запутаются. Например, у сервомотора может быть три разных положения: скажем, левое, верхнее и правое. Положение определяется длиной каждого импульса; импульс длиной 1 мс может обозначать левое положение, 1,5 мс — верхнее, а импульс в 2 мс может переместить его вправо. Так что вероятность ошибки не слишком велика.



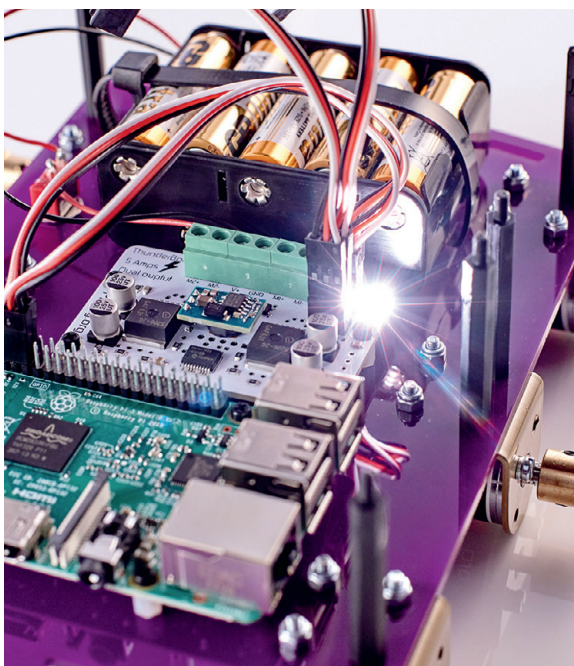


Напугайте коллег, оснастив свой DiddyBorg робокогтем MeArm.

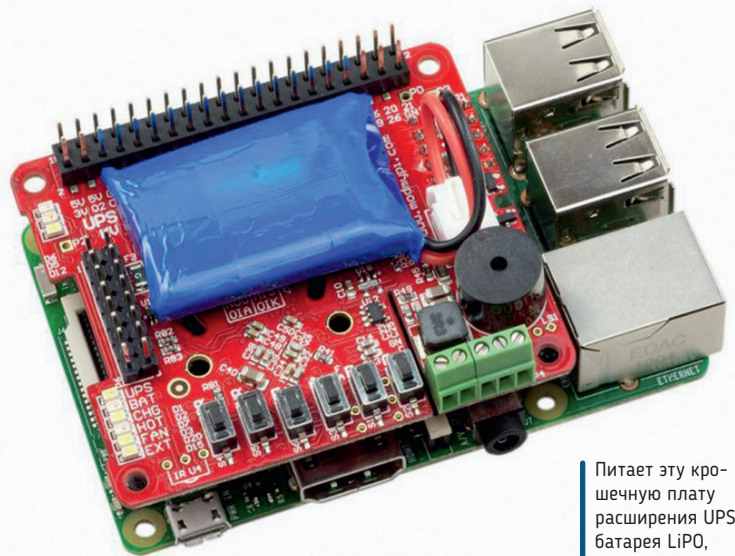
Один из контактов Pi GPIO подключен к аппаратному генератору импульсов, чтобы импульсы генерировались непрерывно, без задержек по причине занятости ОС. Однако иногда одного генератора импульсов недостаточно, и вполне обычным делом является добавление HAT или иной платы расширения (возможно, даже целого микроконтроллера, как, например, Arduino) для работы с несколькими устройствами PWM. Посмотрите, как UltraBorg добивается этого через web-интерфейс, на www.piborg.org/blog/build/ultraborg-build/ultraborgexamples-web-ui.

Улучшенная мощность батарей

Сам Pi может распределять питание на более мелкие устройства через свои контакты GPIO. Но они ограничены до 3,3 В и могут передавать только очень слабый ток. Поэтому через них вряд ли возможно питание моторов с высоким крутящим моментом, ламп или оружия [Ред.: — Это он о чем?], с Pi, подключенным к стандартному банку питания USB. Мы видели, что DiddyBorg получает питание от 10 батарей AA. Однако на данный момент трендом портативного питания определенно являются литиево-ионные полимерные батареи (LiPo). Они более долговечные и имеют ббольшую



У ThunderBorg очень яркой и опционально цветной LED для визуального отображения заряда батарей.



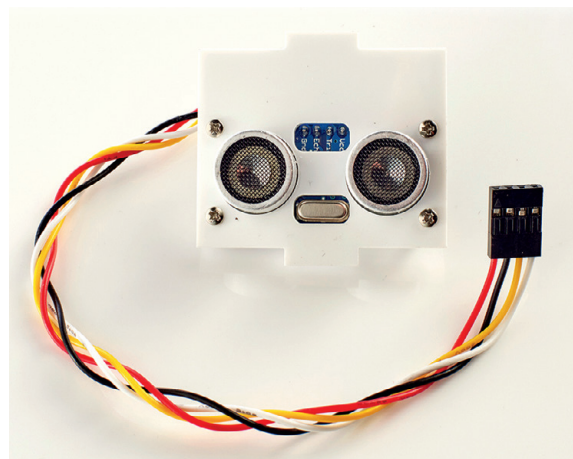
Питает эту крошечную плату расширения UPS батарея LiPo, чтобы ваш Pi не разряжался.

мощность, чем их никель-металгидридные (NiMH) собратья, к тому же дольше держат заряд.

По мере приближения к концу жизненного цикла у батарей наблюдаются скачки напряжения, а некоторые компоненты (не в последнюю очередь сам Pi) весьма к этому чувствительны. Поэтому одной из важных задач, которые выполняет ThunderBorg, подключающий батареи DiddyBorg к Pi, является регулировка питания. Если при включении платы определяется слишком низкое напряжение, она обязательно отключится. Тим Фриберн рассказал нам немного больше об этом стильном создании: «Мы очень гордимся ThunderBorg. Широкий диапазон напряжения на ранних стадиях был ключевой частью дизайна. У ThunderBorg широкий выбор механизмов защиты: например, отключение при низком напряжении, защита от короткого замыкания, защита от перегрева и защита от высокого тока. Благодаря этому сломать его довольно сложно, если только вы правильно подключили питание!»

Ультразвуковые датчики

Помимо использования камеры для оценки окружающей среды, наш робот может уподобиться летучей мыши и использовать для этого ультразвуковые импульсы. Датчик ультразвука, например, HC-SR04, испускает периодические импульсы высокого тона, которые помогают определить по отраженному сигналу близлежащие объекты. Измеряя время между испусканием импульса и обнаружением его отражения и зная, что скорость звука составляет 340 м/с, датчики могут оценить расстояние довольно точно. Наш DiddyBorg поставлялся с набором для монтажа этих датчиков в каждом углу. Такой массив датчиков можно использовать для автоматической параллельной парковки!



DiddyBorg можно оснастить четырьмя ультразвуковыми сенсорами, чтобы он определял препятствия.

Воспарите в небеса

Пс-ст, хочешь подняться в небо вместе с Pi (естественно, в соответствии со местными законами об авиации)?



До сих пор наши ноги/колеса/гусеницы прочно стояли на земле, однако мы полагаем, что наше путешествие в страну роботов надо закончить краткой вылазкой в мир дронов.

Удерживать дрон в воздухе непросто. Каждый мотор должен быть в состоянии реагировать на смену направления почти мгновенно. Определяется это трехосным акселерометром. Нужны и другие инструменты — гироскопы, сервомоторы и, конечно, моторы, чтобы пропеллеры вращались и создавали подъемную силу. Стандартное ядро Linux не предназначено для работы в столь

жестких условиях, и дроны обычно используют ОС реального времени (RTOS), например, NuttX или проприетарную ОС. Однако ядро Linux можно настроить, чтобы оно реагировало не менее быстро. Набор заплат PREEMPT_RT — вероятно, самый популярный способ этого добиться. Этот набор, который долгое время не имел никакого финансирования, теперь является частью RTL Collaborative Project, инициативы с индустриальной поддержкой Linux Foundation. Более подробную информацию см. на <https://wiki.linuxfoundation.org/realtime/rtl/start>.

Erle Robotics (<http://erlerobotics.com>) выпускает Erle Brain — маленькие компьютеры с Linux, способные выполнять все функции, необходимые для удержания дрона в воздухе. Они включают все мыслимые сенсоры и порты расширения. Erle также выпускают PXMini, плату расширения для Raspberry Pi, которая использует Dronocode APM Flight стек. Dronocode — платформа с открытым кодом, которая быстро становится промышленным стандартом. Стек состоит из автопилота PX4, протокола коммуникаций Mavlink, программы QgroundControl и работает на огромном количестве устройств. Читайте больше на www.dronocode.org.

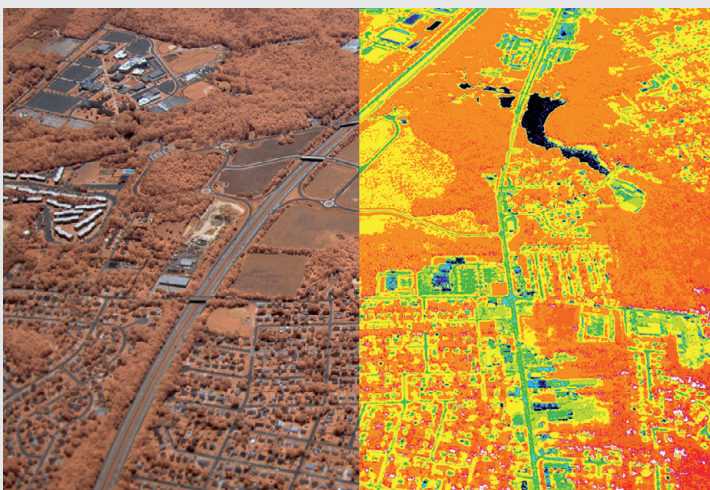
И хотя производители дронов часто предпочитают использовать для управления другую ОС, это не мешает размещению на борту еще одного компьютера с Linux. Что может быть даже интереснее, поскольку позволит все его ресурсы задействовать исключительно на интересные задачи, например, на 360-градусную фотографию через шарнирно-смонтированную камеру.

Кажется, мы лишь слегка коснулись робототехники Pi. Основатель PiBorg, Тимоти Фриберн, говорит нам, что он «работает над роботами на основе Pi для симуляции добычи полезных ископаемых совместно с австралийской горнодобывающей компанией, роботами для проверки воздухозаборного устройства и выходной трубы самолета и роботами для проверки трубопроводов. И этот список очень быстро растет!» Тим также указывает нам на направление некоторых из своих любимых проектов сообщества: Pi In The Sky, The Big Hack и ISS-Above от Лайама Кеннеди [Liam Kennedy] (см. <http://issabove.com/iss-above-and-the-raspberry-pi>). Спасибо, Тим. И спасибо вам, дорогой читатель. А теперь — марш создавать что-нибудь интересное! **LXF**

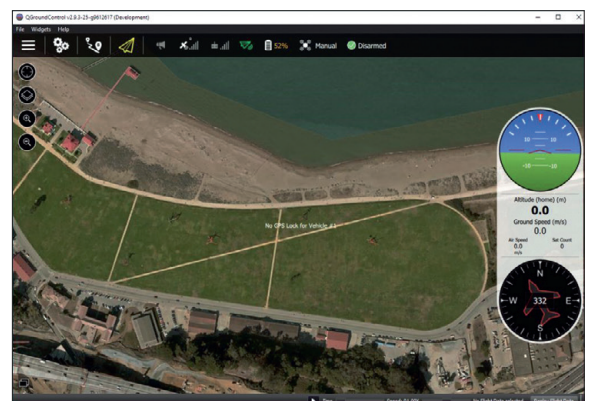
» АЭРОФОТОСЪЕМКА ФЛОРЫ

У камеры Pi NoIR нет инфракрасного фильтра, позволяющего ей «видеть» в темноте, что удобно для приложений видеонаблюдения. Розничный пакет также включает немного странный квадрат из голубого геля. Когда этот голубой квадрат (Roscolux #2007 Storga Blue, как выглядит его полное название) помещается перед объективом, красный и зеленый цвет отфильтровываются, и сенсор смещает инфракрасное излучение в красный спектр, позволяя делать гиперспектральные снимки растений.

Сделанная фотография — это NGB (Near infrared Green Blue — почти инфракрасный Зелено-Голубой) изображение, которое показывает не слишком многое. Но если приложить немного волшебства пост-обработки, можно получить составное изображение NDVI (Near Difference Vegetative Index), которое показывает, хорошо ли в растении идут процессы фотосинтеза, или, грубо говоря, насколько оно здорово. Раньше эта технология требовала специального оборудования, которое монтировалось на самолетах и спутниках. Для серьезных случаев применения оно так и осталось, но посмотрите на проекты инфраграмм на <https://publiclab.org/wiki/raspberry-pi-infragram>, и, в особенности, страницу Мэтью Липпинкотта [Matthew Lippincott] на Flickr на <http://bit.ly/raspberry-pi-infragram>, чтобы узнать, какие еще чудеса можно сотворить со скромным Raspberry Pi.



После пост-обработки изображения NGB (слева) для создания изображения NDVI (справа) мы можем понять, где деревья чувствуют себя хорошо (Изображение CC BY-SA 3.0 Public Lab.)



QGroundControl позволяет отслеживать дрон и составлять карту полетов для важных тайных миссий.

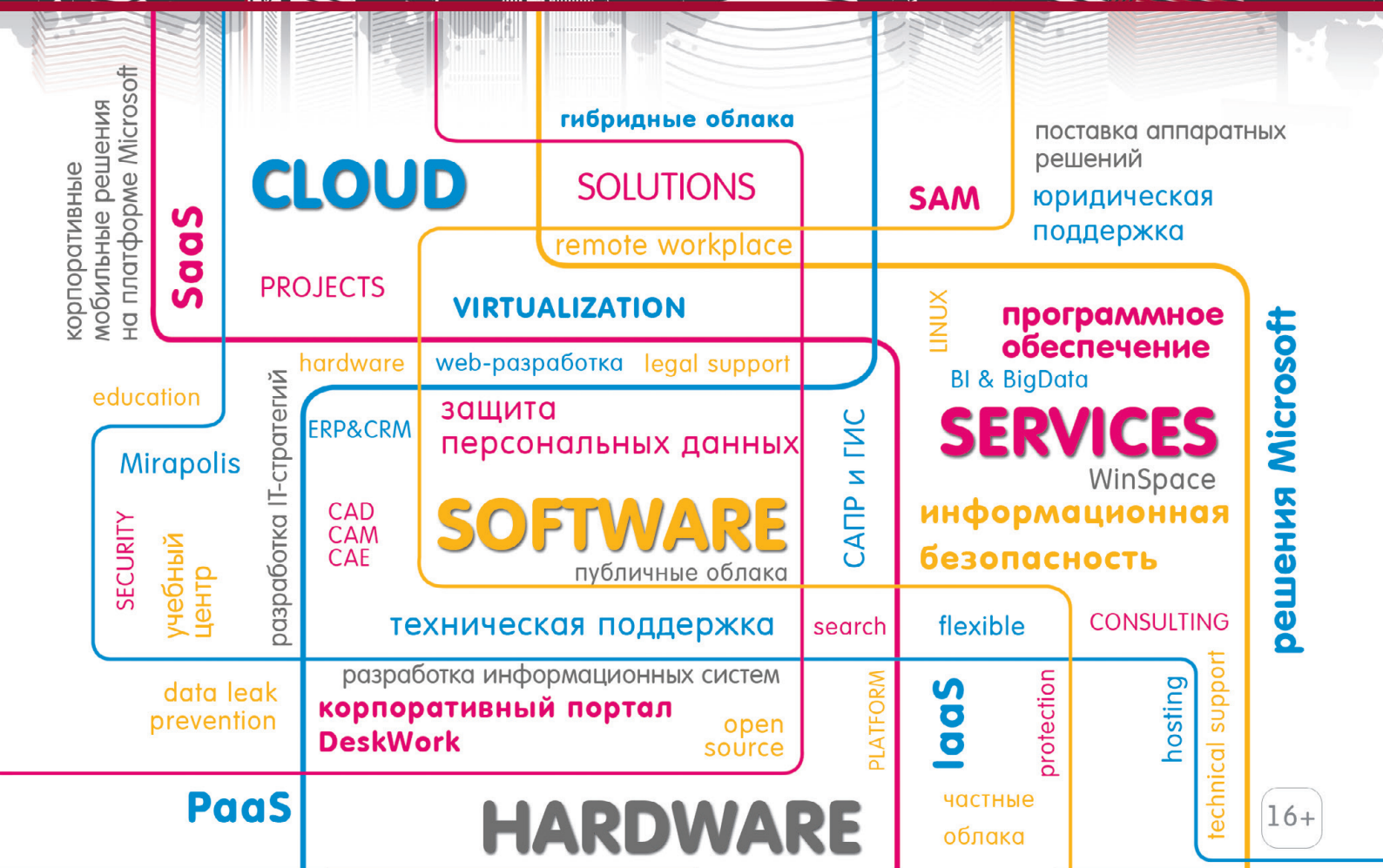
softline®

Cloud Software Hardware Services

20+

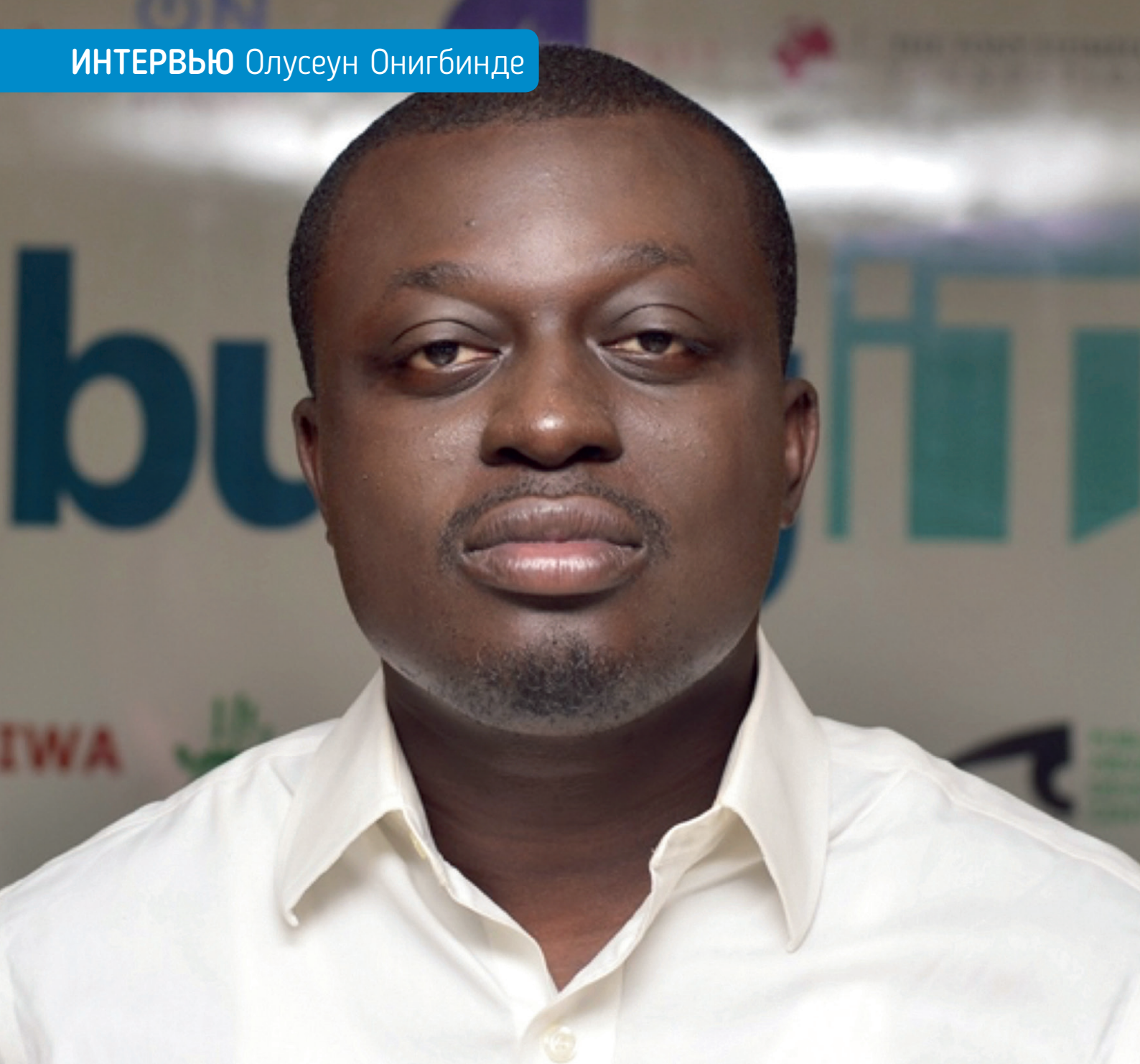
Years in IT

IT-архитектура вашего бизнеса



решения Microsoft

16+



СТАТЬ ПОДОТЧЕТНЫМ

Адам Оксфорд недоумевает: как развить успешную гражданскую техно-инициативу в стране, где лишь один из четверых имеет выход онлайн? Но первопроходец — соучредитель компании BudgIT в Нигерии **Олусеун Онигбинде** нашел способ...

Когда в 2016 г. Билла Гейтса [Bill Gates] пригласили в Южную Африку прочитать 14-ю ежегодную лекцию Нельсона Манделы в Преторийском Университете, он долго и красноречиво говорил о потенциале технологий, касающихся межконтинентальных вопросов: энергии, создания рабочих мест, финансовой интеграции, сельского хозяйства и здравоохранения.

Разумеется, он говорил о работе Фонда Билла и Мелинды Гейтс (BMGF) по всей Африке и о ряде проектов, виденных им во время путешествия. Также он отметил нескольких личностей: Нельсона Манделу [Nelson Mandela] и его вдову Грасу Машель [Graça Machel], пригласившую его в Университет, Стива Джобса [Steve Jobs] и Марка Цукерберга [Mark Zuckerberg]. А также одного общественного предпринимателя, который сумел изменить то, как Нигерия управляет своими деньгами.

Олусеун Онигбинде [Oluseun Onigbinde] — 32-летний соучредитель компании BudgIT (www.yourbudgit.com), гражданской организации, которая через открытые порталы данных делает государственные расходы подотчетными. Это началось в 2011 г. с простого онлайн-инструмента для отслеживания расходов национального бюджета, а теперь обеспечивает прозрачность и подотчетность целому ряду национальных и локальных планов правительства.

Через год после выступления Билла Гейтса BudgIT получил грант в размере \$3 миллиона от BMFG и фонда Omidyar, на некоторое время обеспечив свое будущее.

Последний инструмент организации — Траска, помогает гражданам отслеживать обещанные капиталоемкие проекты, такие как новые школы и клиники, и позволяет им сообщать о подозрительных проблемах, связанных с мошенничеством и нерациональным использованием средств.

В Африке сейчас активно развивается гражданская техническая сфера, и Лагос стал Меккой для западноафриканских стартапов и инвесторов.

Но еще в 2011 г. всё было иначе. По оценкам Всемирного банка, интернет-покрытие в Нигерии составляло всего 28% (сегодня это около 40%), и государство сильно сопротивлялось проверкам (оно по-прежнему насчитывает всего 17 из 100 в международном индексе открытости бюджета).

Linux Format поболтал с Олусеуном, чтобы узнать, как идет проект и какой был достигнут прогресс.

Linux Format: Расскажите нам немного об основании BudgIT. Как это произошло?

Олусеун Онигбинде: В то время я был обычным банковским работником в First Bank, в Лагосе. В 2010 г. co-Creation Hub (<http://www.cchubnigeria.com>) только что открылся, и был раз-

от фонда «Ашока [Ashoka]», чтобы освоиться в новом положении.

LXF: Такие виды гражданских технических проектов тогда еще были новыми во всем мире, но в Африке особенно. Даже кенийскому iHub было всего лишь около года. Это было довольно маленькое сообщество на континенте, так что же это было?

ОО: Все хотели, чтобы мы быстро изменили ситуацию и продвигались в разные места, потому что такой вид проектов по отчетности необходим везде. Но мы обнаружили, что объем работы даже в одной Нигерии огромен. Нигерия велика — около 200 миллионов человек, а государственный бюджет — 9,15 триллионов найра

ПОЧЕМУ BUDGIT СТАЛ УСПЕШНЫМ

«Все хотели, чтобы мы быстро изменили ситуацию и продвигались в разные места, потому что такой вид проектов по отчетности необходим везде.»

работан для поддержки и поощрения социальных инноваций и проектов, способных применять технологии, помогающие преобразовать организацию управления.

В co-Creation Hub есть много гражданских узкоспециализированных рабочих групп программистов, поэтому в 2011 г. я и мой друг Джозеф Агунбиаде [Joseph Agunbiade] выступили с идеей BudgIT. Наша основная цель состояла в исследовании работы общественных проектов и побуждении людей задавать вопросы насчет расходов.

После хакатона нас поощрили продолжать BudgIT и заниматься им полный рабочий день. Для меня это был пугающий шаг, ведь в банке я преуспевал. Но, к счастью, я получил стипендию

(около \$25 триллионов). Здесь еще многое предстоит сделать.

Но самое главное, мы обнаружили, что вопросы отчетности разные в каждой стране. Если нечто работает здесь, это отнюдь не значит, что оно будет работать в Гане, Кении или Уганде.

Поэтому мы приняли стратегическое решение придерживаться Нигерии — по крайней мере, на первое десятилетие. Правда, у нас есть офис в Сьерра-Леоне в качестве пилотного проекта панафриканской экспансии.

LXF: В стране, где почти никто не имел доступа к Интернету и интерес к отчетности правительства был низок, как вам удалось убедить людей использовать BudgIT?

ОО: Сначала продать идею людям было трудно. Первоначальный отклик не был особенно бурным. Однако нас сильно подтолкнул вперед большой протест 2012 г. До тех пор правительство субсидировало местные цены на топливо из денег, полученных за добычу нефти. Цены были примерно \$0,40 (по курсу 2012 г.) за литр, и большинство людей использовали небольшие генераторы, работающие на дешевом дизельном топливе [Ред.: — По данным *The Guardian*, национальная сеть тогда обеспечивала всего 40 Вт энергии на каждого гражданина, а ежегодная субсидия была около \$6 млрд].

Но потом президент Гудлак Джонатан [Goodluck Jonathan] принял решение в один день снять субсидию: понадобились деньги

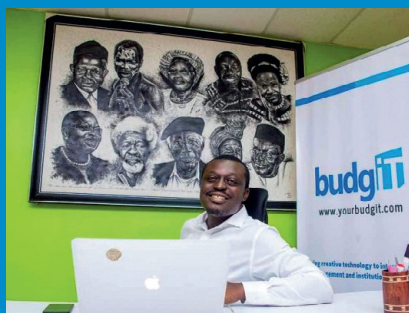
» ОЛУСЕУН ОНИГБИНДЕ

Проживает: Лагос, Нигерия.

Место работы: Соучредитель и генеральный директор BudgIT (www.yourbudgit.com).

Образование: Бакалавр по электротехнике/электронике, Федеральный университет сельского хозяйства, Абеокута; и Стэнфордский Университет, Программа для управляющих в социальном предпринимательстве.

Награды: Стипендия Ashoka для бизнеса мирового масштаба, World Summit Youth Award, стипендиат Knight ICFJ International Journalism.



на капитальные затраты. Возник мощный протест, и все вдруг заинтересовались, куда же идут деньги.

Нигерийцы начали задавать вопросы о госзатратах. Люди придерживались мнения, что если правительство хочет поднять цены на нефть, то они имеют право видеть свидетельства, оправдывающие колоссальные издержки управления.

Поэтому люди начали изучать бюджеты и потребовали их публикации. Госзатраты и коррупция всегда были проблемой для страны, но тут общественность пожелала убедиться, что новые планы расходов действительно будут эффективными.

Нам удалось оседлать гребень этой волны общественных интересов. Гражданская активность, прозрачность и открытые данные были в то время новостью. Появилось много организаций, которые хотели данных и анализа,

и мы заполнили этот пробел. В социальных сетях были кампании по типу «следи за деньгами», но то, что привнесли мы, прояснило, как взаимодействовать с данными в принципе и как вовлекать миллионную аудиторию.

LXF: Звучит потрясающе — быть в эпицентре такого движения! Каким образом BudgIT развивался с годами?

ОО: Мы постоянно пересматриваем свое мышление о бюджетах и подотчетности, бросая вызов своему пониманию данных и аудитории. Мы охватили где-то от 2 до 3 миллионов человек и получили много отзывов, изменивших нашу деятельность.

Мы посмотрели, как нам добраться до людей на местном уровне. Например, Traska — это предоставление власти непосредственно сооб-

КАК ПРАВИТЕЛЬСТВО ВИДИТ BUDGIT

«Иногда вы заклятый враг, иногда им нравится ваша работа и то, как вы помогаете гражданам осваивать трудно осмысливаемую информацию.»

» ТЕХНОЛОГИИ В ЛАГОСЕ

Нигерийская техническая сцена сосредоточена в портовом городе Лагос, обширном и хаотичном месте — самом крупном городе страны, насчитывающем 12 миллионов жителей в зоне большого метро. Если говорить точнее, эпицентр деятельности стартапов находится в районе Яба [Yaba], где, между прочим, располагается и co-Creation Hub.

Интерес к региону стремительно взлетел. В прошлом году его посетили компании 500Startups и Y Combinator, а организация VC4Africa считает, что нигерийские предприниматели привлекали больше

зарубежных инвестиций, чем кто-либо другой на континенте.

Частично это благодаря фирме программного обеспечения Andela, которая получила обильные инвестиции в размере \$24 млн от фонда Chan Zuckerberg Initiative в 2016 г. и еще \$40 млн в раунде финансирования C Series в прошлом году. Andela также готовит местных разработчиков и предоставляет услуги аутсорсинга для зарубежных фирм. Она расширилась за пределы Нигерии в Кении, Гане и Уганде, и имеет офисы в Нью-Йорке и Сан-Франциско.

ществам. В настоящее время мы отслеживаем порядка 40 инфраструктурных проектов и сохраняем данные о них открытыми для сообществ на местах.

LXF: Правительство Нигерии не всегда славилось терпимостью к инакомыслию и гражданской активности. Мы в Великобритании видели открытые организации данных, такие как MySociety, работающие с госслужбой, но какова реакция на BudgIT?

ОО: Иногда отношения с правительством прекрасны, но иногда они не любят нас совсем. Честно сказать, это вроде суровой мужской дружбы. Иногда вы заклятый враг, иногда им нравится ваша работа и то, как вы помогаете гражданам осваивать трудно осмысливаемую информацию.

Что касается федерального правительства, то от него довольно легко потребовать прозрачности. А иногда и трудно, потому что вскрывается неприятная правда и официальные власти хотят придержать данные; но общая тенденция — это открытость.

На уровне местной власти состояние совершенно иное. Они просто не привыкли, что им кто-то может задавать вопросы. Гражданское общество на этом уровне действительно слабое, и имеется ореол безнаказанности. Таким образом, большая часть работы является постепенной, необходимо много образовательской деятельности. Правительству надо серьезно обучаться подобным инструментам: какая от них польза и как с ними взаимодействовать.

Но урок, который мы извлекли как внутри правительства, так и от конечных пользователей, заключается в том, что люди действительно рады публикуемым нами данным и предлагаемой нами информации, если мы сумеем показать, что она работает.



LXF: Каков был самый большой успех за последние семь лет?

ОО: Национальное собрание. Без сомнения. Кульминацией было, когда мы заставили их раскрыть отчеты о затратах; но это был долгий процесс. Мы проводили кампанию в течение двух лет, инвестировали в нее массу ресурсов, но ничего о Собрании и его расходах не публиковалось. Они просто скрывали свои бюджеты, и нам приходилось задавать всё тот же вопрос снова и снова. Получив, наконец, эти данные, я испытал огромную радость.

LXF: Как вы думаете, почему никто раньше не боролся за эти данные?

ОО: Есть много причин. Бедность и плохое образование... многие вещи удерживают людей. Вы боитесь правительства, вы просто верите правительству или вы не знаете, что спросить. Вот почему так здорово, что мы вовлекаем общественность. Людям нравится, когда мы показываем им наши инструменты и обеспечиваем немного большее осознание того, как работает правительство. Но мы должны сделать данные пригодными для использования: просто публиковать их — недостаточно. Люди должны уметь понимать их и то, каким образом данные способны принести им пользу. Это должны быть работающие данные. Итак, когда мы выкладываем любые данные, то всегда следим за тем, чтобы было ясно, чем они полезны обычным людям.

LXF: Каково это — быть упомянутым в речи Билла Гейтса?

ОО: Да, это было круто. Это свидетельство важности нашей работы. Мы продолжаем идти дальше.

LXF: Какие задачи стоят перед BudgIT?

ОО: Главная проблема в Нигерии в целом то, что было очень много надежд, связанных с избранием нынешнего правительства [Ред.: — Президент Мухаммед Бухари [Muhammad Buhari] был избран в 2015 г., с антикоррупционной платформой],



Олусеун говорит, что BudgIT должен вникать в данные, а не просто публиковать их.

» АФРИКА ОТКРЫТОГО КОДА

Проведя много времени среди технических компаний и инициатив в Африке, невозможно не заразиться постоянно растущим энтузиазмом по поводу потенциала технологий на континенте. Будь то проблемы государственной отчетности, правовых норм, ведения бизнеса или образования — видение, выражаемое активистами, сотрудниками корпораций или политиками, заключается в том, что цифровые технологии перескажут столетия инвестиций в инфраструктуру, как сделали мобильные телефоны в сфере коммуникаций.

Некоторые из них типичны для техно-евангелизма Кремниевой долины, и циники могут справедливо отметить, что многие из вариантов, предлагаемых для решения социальных проблем на всем континенте, также дают мегакорпорациям США удобные способы нахождения новых клиентов и потоков прибыли. Эта позиция подытожена в названии новой книги Ноама Коэна [Noam Cohen] «Знай-всё-всё». Так или иначе, без дальних обобщений, импульс влияния технологий был положительным.

В 2015 г. Всемирный банк посчитал количество технохабов, зарегистрированных на континенте. Это было отличное упражнение, потому что площадки наподобие CoCreation Lab сыграли жизненно важную роль в фокусировании сообществ на местных пробле-

но люди очень разочаровались. Возникло ощущение, что «все политики одинаковы».

Наша задача — предоставить правительству стимул к переменам. Это немного похоже на воспитание ребенка. Это непослушное правительство — вы кормите его, одеваете и воспитываете. И можно было бы ожидать, что оно будет использовать преимущества прозрачности и подотчетности для себя; но нам следует быть гораздо бдительнее. Мы должны общаться с ними и продолжать разговор о ведении открытой правительственной деятельности, или они не изменятся. Но мы бдительные существа.

LXF: А какие могут быть проблемы с технической точки зрения?

ОО: В нашей работе мы используем самые разные технологии. Мы некоммерческая организация, и должны использовать всё, что нам доступно. Всё, что мы строим, мы публикуем как открытый код, но и применяем много продуктов вроде Slack и Google Docs. Наша главная цель — быть сильнее в том, как мы общаемся. Мы не можем просто собирать, обрабатывать и публиковать данные.

мах гражданской и деловой жизни и вывели их на людей с навыками их решения (как, например, команда, стоящая за BudgIT). Это проблематично, потому что определение «технохаба» довольно расплывчато и не гарантирует наличия какой-либо полезной деятельности.

Тем не менее, Всемирный банк справедливо отмечает, что «технологические кластеры создают эффект снежного кома». В 2015 г. было 117 хабов, а чуть менее двух лет спустя GSMA подсчитала 314, «число, которое продолжает расти еженедельно».

Многие из этих хабов и стартапы, которые они выращивают, потерпят неудачу. Но их влияние ощутимо. В августе 2013 г. в африканских странах было зарегистрировано только 4537 учетных записей GitHub, и почти все они базировались в Южной Африке, Египте и Кении. Общее количество учетных записей GitHub в то время было 3850000, так что африканцы составляли всего 0,12% от общей численности разработчиков с открытым исходным кодом, использующих этот вездесущий инструмент.

Ныне насчитывается 24 миллиона учетных записей GitHub, т.е. общее количество увеличилось шестеро. Но 486000 из этих пользователей сейчас находятся в Африке, и это в сотню с лишним раз больше, чем в 2013 г. И их число продолжает расти.

Мы должны быть более творческими, быть быстрее в том, что мы делаем, сосредоточиться на повествовании.

Мы используем Digital Ocean для хостинга, который при необходимости позволяет нам масштабироваться — всегда бывает мощный всплеск интереса в течение рабочего времени. Тем не менее, всегда есть проблемы с безопасностью. Не всем нравится то, что мы делаем, и в прошлом мы целыми ночами отбивались от DDoS-атак. Это расстраивает меня, но не сильно.

LXF: Какие еще технические инициативы в области гражданского общества Нигерии вас волнуют?

ОО: Я в восторге от организации Connected Development и их работы по мобилизации граждан в местных сообществах на требование эффективных действий. Мне также нравится начинающий стартап, известный как Gavel. Gavel работает над улучшением доступа к правосудию, особенно для лиц, находящихся в неблагоприятном положении. Я доволен тем, что YIAGA Africa делает с голосованием по основанному на данных подходу к целостности выборов. **LXF**

АЛЬТ И «БАЗАЛЬТ»

Игорь Штомпель выяснил у **Андрея Черепанова**,
как собираются российские дистрибутивы.

Андрей Черепанов родился 22 декабря 1974 г. в Красноярске. В 1998 г. закончил с отличием Сибирскую аэрокосмическую академию по специальности «Экономика и менеджмент». В настоящее время работает в компании ООО «Базальт СПО» начальником отдела поддержки и администрирования. Он один из ведущих локализаторов KDE на русский язык.

Linux Format: Расскажите, чем вы занимаетесь в «Базальт СПО».

Андрей Черепанов: Я возглавляю отдел технической поддержки и администрирования. Хотя функции отдела намного шире ответов на вопросы и поддержания инфраструктуры: это и тестирование как собственных дистрибутивов, так и продуктов сторонних производителей, обеспечение качества и доработка пакетов в репозитории, написание документации и руководств.

Как начальник отдела, я ставлю задачи и контролирую выполнение, общаюсь с заказчиками. Как мейнтейнер ALT Linux Team, собираю пакеты (как свои, так и чиню бесхозные).

LXF: В чем основные новшества Восьмой платформы Альт Линукс?

АЧ: Давайте я попробую разрешить небольшую путаницу, которая сложилась в головах пользователей. Итак, долгое время работала компания ООО «Альт Линукс», которая выпускала дистрибутивы Альт Линукс. И, соответственно, владела торговой маркой Альт Линукс. По корпоративным причинам руководство и разработчики, ранее работавшие в ООО «Альт Линукс», перешли в новую компанию — ООО «Базальт СПО». И уже в этой компании выпустили дистрибутивы под торговой маркой Альт (ведь торговая марка на Альт Линукс остались в ООО «Альт Линукс»). Немного особняком стоит торговая марка сертифицированного дистрибутива Альт Линукс СПТ, которая принадлежит ООО «СПТ», с которой и Альт Линукс, и Базальт СПО делали дистрибутивы.



Microsoft Exchange в виде SOGo, решения по виртуализации на базе PVE и OpenStack.

Для рабочих станций было принято решение о разделении сборки обычного Firefox и сборки Firefox ESR с длительной поддержкой, разнообразные решения по работе в корпоративных доменах (как правило, Active Directory, но и FreeIPA поддерживается).

LXF: Как происходит создание новых дистрибутивов Альт на базе репозитория Sisyphus? Опишите, пожалуйста, основные этапы.

АЧ: Дистрибутивы Альта сейчас собираются с помощью наших инструментов *mkimage-profiles* или устаревшего *mkimage-profiles-desktop*. Фактически это набор скриптов и правил *Makefile*, которые при наличии репозитория, клонированного Git-репозитория *mkimage-profiles*, после за-

правило на базе существующего, добавить нужные фичи и пакеты, и... voilà, свой дистрибутив готов. Правда, для нескудных обоев нужно сделать свой <http://altlinux.org/branding>, но это тоже не rocket science. Доказано разрабатываемыми сторонними дистрибутивами.

LXF: Что является характерными особенностями систем Альт?

АЧ: Я бы назвал следующие: независимость от иных репозиториях (Альт основывается на собственном репозитории Sisyphus, одном из крупнейших репозиториях свободного программного обеспечения в мире, со своей системой сборки и контроля целостности и разными интересными технологиями).

Например, каждый исходный пакет (а их сейчас в репозитории почти 18 тысяч) автоматически сохраняется в виде репозитория Git с историей всех изменений. Да и сами пакеты проще собирать непосредственно из апстримных репозиториях Git, добавляя пару файлов.

Контроль целостности не позволяет при установке или обновлении получить неудовлетворяемые зависимости (unmet). Большое внимание уделяется безопасности: не просто собирать свежие версии с исправлением свежесобранной уязвимости, а сделать так, чтобы потенциальные уязвимости не появились или не нанесли урон. Например, запуск многих служб в изолированном окружении (chroot).

Хочется отметить модульную платформу для конфигурирования *Alterator*, на которой сделана как программа установки, так и Центр управления системой (с интерфейсом как на *Qt*, так и с веб-интерфейсом). Аналогична *YaST* и *drakconf*. В репозитории под нее есть более 100 модулей по настройке.

Интересующиеся могут почитать про остальные особенности на Wiki сообщества: <https://www.altlinux.org/Features>.

О ВНИМАНИИ К БЕЗОПАСНОСТИ

«Не просто собирать свежие версии с исправлением свежесобранной ошибки, а сделать так, чтобы потенциальные уязвимости не нанесли урон.»

То есть дистрибутивы, начиная с 8.0, выпущенные Базальт СПО, называются просто Альт: Альт Сервер, Альт Рабочая станция, Альт Образование. Сообщество сопровождающих пакеты — ALT Linux Team — так и продолжает называться.

Возвращаясь к Восьмой платформе, которая была целиком выпущена уже силами «Базальт СПО», отмечу, что основным вектором стала работа над инфраструктурными решениями, востребованными в компаниях. Это и предложение заказчикам инфраструктуры доменов Active Directory (на базе Samba) и FreeIPA, и решения по замене

пуска команды типа *make education.iso* позволяют получить готовый файл ISO с дистрибутивом. В правилах указываются особенности (+efi +systemd) и группы или отдельные пакеты, устанавливаемые на разных этапах (@\$(call add,MAIN_PACKAGES,kumir2 rujel)). Более подробно об этом можно почитать на сайте <https://www.altlinux.org/M-p>.

При этом в одном репозитории есть профили всех выпущенных дистрибутивов Альта (кроме Альт Сервера, который собирался на *mkimage-profiles-desktop*). Достаточно создать свое





LXF: Каково назначение *Alterator*? Что нового появилось в нем в Восьмой платформе? Если возможно, раскройте историю его появления.

AЧ: *Alterator* служит для настройки системы. Сделали его так, чтобы можно было расширять, дописывая модули (как *YaST* или *drakconf*). Так как модули живут в рамках своих жизненных циклов, то особой привязки к платформе нет. Помню, что появилось создание доменов Active Directory и FreeIPA в *alterator-net-domain*. Хотя исправлений и изменений в разных модулях *Alterator* масса.

LXF: Будет ли реализован в дистрибутивах Альт «Центр приложений»? Какова будет его

архитектура, на основе каких приложений он может быть реализован?

AЧ: Обязательно будет! Очень многие пользователи просят его. Только вот срок появления его на свет точно назвать не могу. На сегодняшний день незачем придумывать свои велосипеды в эпоху устоявшейся архитектуры Appstream и описаний в файлах *metainfo.xml*, которые доступны уже для многих приложений и есть шикарная база в Fedora. Собственно, потихоньку база приложений собирается в пакете *appstream-ata*, осталось только сделать интерфейс для просмотра базы и установки приложений, интегрированный в него.

Конечно, можно было бы использовать *gnome-software-center*, но сборка и адаптация нужного для него PackageKit под *apt-rpm* — довольно трудоемкая задача. Хотя далеко не всем пользователям сред, помимо Gnome, может понравиться специфическое гномовское юзабилити.

LXF: Что представляет собой процесс сборки пакета, какие инструменты используете? В чем его особенности? Каковы стратегии сборки («с нуля» и т.п.)?

AЧ: Сборка пакета фактически сводится к клонированию апстримного репозитория Git или созданию такого репозитория на основе архива с исходным кодом, добавлению файла спецификации [sрес] и сборке с помощью нашего средства Gear в изолированном окружении (Hasher). Изолированное окружение гарантирует выставление правильных зависимостей как на сборку, так и в получающихся пакетах, то есть обеспечивает воспроизводимую сборку. При этом локально не надо держать пакеты разработчика для каждого пакета: при сборке будет создан специальный *chroot*, в который будут поставлены все указанные в файле *срес* зависимости сборки. Один минус у такого решения — должно быть под рукой зеркало репозитория или хороший доступ к репозиторию в Интернете. После того, как всё локально собирается, тег отправляется на наш сервер Git — gitery.altlinux.org, доступ к которому имеют все мейнтейнеры, и по SSH дается команда на сборку пакета, включающая имя репозитория и тег. Наш репозиторий Git можно посмотреть через веб-интерфейс git.altlinux.org.

Сборка в Альт достаточно строгая: пакет должен соответствовать FHS, не должно появляться неудовлетворенных зависимостей [unmets] при автоматическом поиске зависимостей, не должно быть недолинкованных библиотек и bad elf symbols. Для этого на этапе сборки в сборочнице применяются множественные проверки.

Еще одной особенностью сборки являются зависимости *setversion*, которые фактически фиксируют хэш ABI библиотек. И при его смене требуется пересобрать зависимые пакеты. Это эффективнее, чем, как в Fedora, чохом периодически пересобирать всю пакетную базу.

LXF: Вы рассказали о сборке пакетов, а как организовано управление пакетами в дистрибутивах Альт? Какое программное обеспечение используется для этого?

AЧ: Используется *apt* поверх RPM. Соответственно, ставить можно как из командной строки, используя привычный пользователям Debian *apt-get*, так и через графический интерфейс *Synaptic*.

LXF: Расскажите о сборке проприетарных пакетов. В чем ее особенности?

AЧ: Если кому-то нужно собрать двоичный пакет, который будет устанавливаться в наши дистрибутивы, но при этом он не хочет его помещать в репозиторий, то он может собрать его локально в Hasher на установленном дистрибутиве Альт и дальше передавать собранный пакет.

Так что особенностей, помимо организационных и юридических, нет никаких.

LXF: В репозиториях дистрибутивов Альт Восьмой платформы имеются пакеты, содержащие в названии “-preinstall”. Например: “yandex-browser-preinstall”, “i586-skype-preinstall.32bit”, “i586-teamviewer-preinstall.32bit” и другие. В чем заключается их назначение?

AЧ: Подготовить пакетную базу для установки стороннего пакета. То есть при включении таких пакетов в дистрибутив нам не потребуется для установки того же *Skype* что-то дотягивать из репозитория. Иногда эти пакеты предоставляют хитрые зависимости, которые есть в этих популярных пакетах, но у нас называются по-другому.

LXF: Планируется ли включение в репозитории набирающего популярность редактора *Atom* и *Sublime Text*? *Visual Studio Code*, например, уже был добавлен Виталием Липатовым (<http://sisyphus.ru/ru/srpm/Sisyphus/code/>)

AЧ: *Visual Studio Code* — это всего лишь текстовый редактор. Как *Kate* по сравнению с *KDevelop*. Работы по сборке *Electron* и модных приложений на его основе идут, но особой спешки нет, так как любой желающий притащить себе еще одну оболочку с браузером может сделать это и в собранном виде с официального сайта.

LXF: Какие дистрибутивы были выпущены на основе Восьмой платформы? На решение каких задач они ориентированы?

AЧ: Альт Сервер предназначен для серверных задач: от роли контроллера домена до шлюза, сервера баз данных или ноды в облаке.

Альт Рабочая станция (есть варианты на *Mate* и на *KDE5*) предназначена для корпоративных персональных компьютеров (хотя можно использовать и дома).

Альт Образование (преемник Альт Линукс Школьный) предназначен для образовательных учреждений и просто для обучающихся. Такой ребрендинг был сделан потому, что пользователи не очень любят ассоциации со школой. Да и многие заказчики (от детских садов до ВУЗов) требуют другого названия. В этом дистрибутиве много образовательных программ и средств разработки.

LXF: Поддерживается ли работа с *Active Directory*? Работает ли миграция с домена *Windows* на домен Альт?

AЧ: Работает.

LXF: В чем заключаются возможности системы управления виртуализацией в системах Альт? Каковы планы развития?

AЧ: *PVE*, *OpenStack*.

LXF: Для допуска операционных систем в государственные учреждения требуется наличие сертификации ФСТЭК России. Какие операционные системы Альт прошли такую сертификацию?

AЧ: Скорее, для тех мест, где работают с персональными данными (желающие могут почитать

на досуге Федеральный закон РФ от 27 июля 2006 г. № 152-ФЗ «О персональных данных»), требуется провести аттестацию. Как, впрочем, и для мест, где работают с конфиденциальной информацией. И эту аттестацию проще проходить с сертифицированными средствами защиты информации. Для этого у нас есть Альт Линукс СПТ 7.0, сертификация которого недавно была завершена. С этим дистрибутивом можно смело аттестовывать информационные системы персональных данных (ИСПДн) по класс 1К включительно.

LXF: Каковы сроки поддержки дистрибутивов?

AЧ: 3 года для обычных дистрибутивов, 5 лет — для сертифицированных.

LXF: Поговорим о безопасности. Есть ли свои разработки на уровне архитектуры систем Альт?

AЧ: Да.

LXF: Как организован процесс исправления уязвимостей?

AЧ: Наши сотрудники подписаны на закрытые списки рассылки ранних сообщений об уязвимостях, и поэтому мы можем заранее подготовить исправления.

LXF: Расскажите о работе над поддержкой российских процессоров Эльбрус, Байкал. Ведется ли она?

AЧ: Да, конечно. У нас стоит и десктоп, и сервер на Эльбрусе. На сегодня собрано более 2200 пакетов, что позволяет поставить на Эльбрус полноценный рабочий стол на *Xfce* или сделать контроллер домена *Active Directory* на *Samba-DC*. К концу 2018 г. мы ожидаем, что появится дистрибутив для Эльбруса.

Что касается Байкал Электроникс, то там есть два направления: *MIPS* (Байкал-Т1) и *AArch64* (он же *ARMv8*, он же Байкал-М). 32-битный *MIPS*

О ДИСТРИБУТИВЕ АЛЬТ ЛИНУКС СПТ 7.0 «С этим дистрибутивом можно смело аттестовывать информационные системы персональных данных по класс 1К включительно.»

не особо интересен из-за производительности, поэтому мы им не занимаемся. А для *AArch64* у нас есть готовый репозиторий, собираемый на 64-битных *ARM*-серверах, но вот Байкал-М пока неясно когда появится. Также у нас есть репозиторий для *ARMv7*.

Есть интересные предложения по портированию Альта на 64-битный *MIPS* и на *мейнфреймы*.

LXF: Включено ли семейство операционных систем Альт в Реестр российских программ? Какие применяются платформы?

AЧ: Да. Пока — *Intel*, хотя у нас есть решения и на Эльбрусе, и на *AArch64*.

LXF: С чего следует начинать разработку, который хочет внести вклад в развитие репозитория *Sisyphus*?

AЧ: С прочтения <https://www.altlinux.org/Join>. Помимо желания, потребуется создать два ключа (*SSH* и *PGP*) и указать ментора. Ментора легко найти среди существующих *мейнтейнеров*. А уж дальше всё зависит от самого человека, поскольку после вступления в *ALT Linux Team* он может собирать в репозиторий всё что угодно (при условии, что это пропустит сборочница). Впрочем, многие пользователи собирают пакеты и без участия в *Team*.

LXF: Какое программное обеспечение предпочитаете? Для повседневной работы (браузер, почтовых клиент и т.п.), для разработки (среды, редакторы и т.п.)?

AЧ: Уже лет 17 как последовательно люблю *KDE*. В качестве браузера — *Chromium: Firefox* мне так и не зашел. После ужасного извращения, сделанного с *KMail* в эпоху *KDE4*, мне пришлось переходить на *Thunderbird*. В основном разработка идет по *SSH* на сервере, поэтому — *Bash*, *Vim* и всё многообразии консольных утилит. Если надо отдельно графический проект сделать (тот же *packageinstall*), то *Qt Creator*.

Для отслеживания задач и заметок на сервере использую *Task Warrior*, на десктопе — *Basket* и *KWrite*.

LXF: Есть ли аппаратные предпочтения?

AЧ: Не *HP* и не *AMD*. Из ноутбуков люблю *ASUS*, но вот они слишком любят новое железо, поддержка которого реализуется только в последних ядрах.

LXF: В каких мероприятиях, конференциях принимаете участие? В каких планируете выступать в 2018 г.?

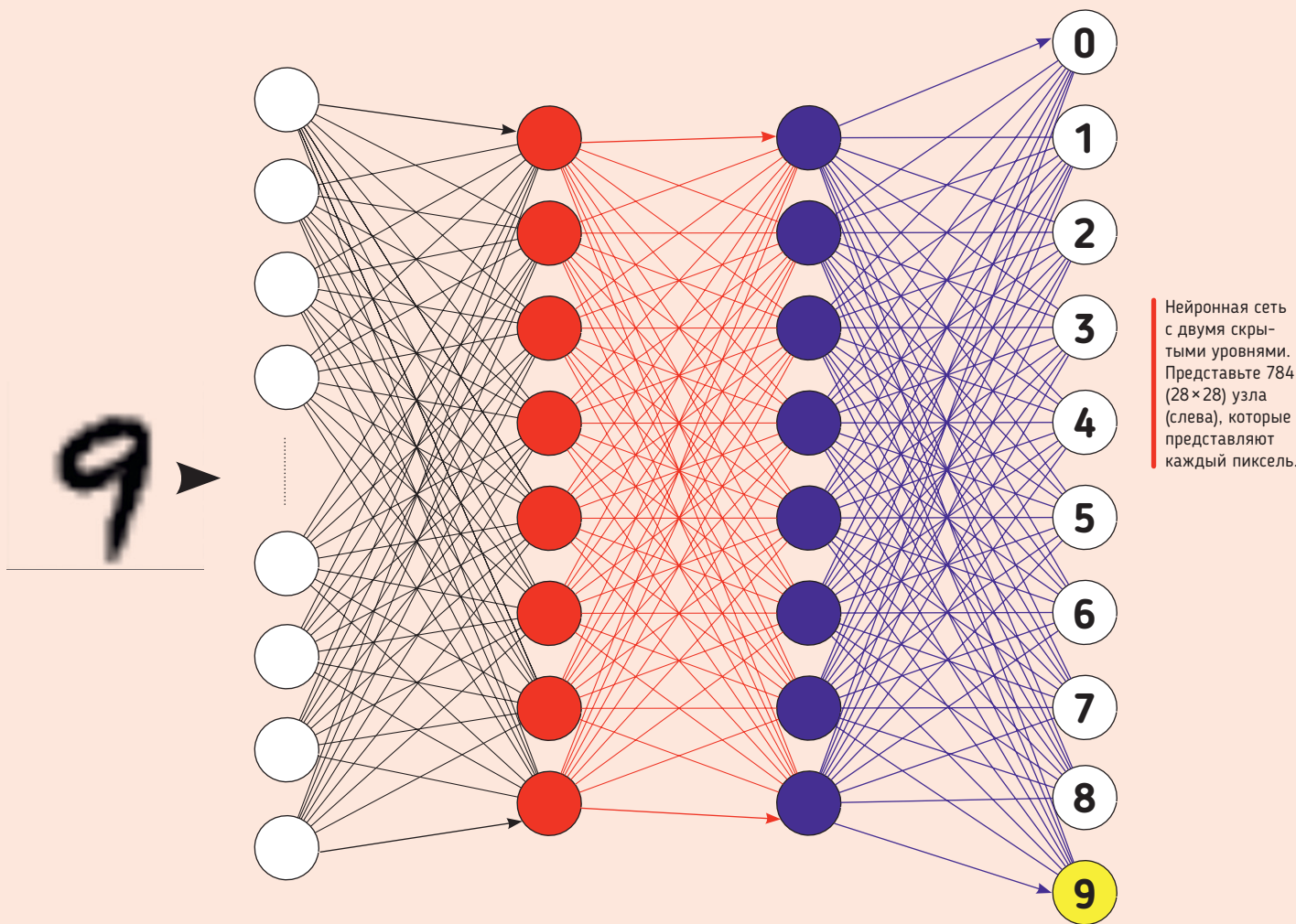
AЧ: Традиционно сотрудники Базальт СПО ездят на ежегодную конференцию разработчиков свободного программного обеспечения (последняя проходила в Калуге 22–24 сентября 2017 г.) и зимнюю конференцию в Переславле-Залесском «Свободное программное обеспечение в высшей школе» в конце января. Я был на обеих конференциях в качестве слушателя.

LXF: Есть ли у вас тематика, которую хотелось бы осветить? Или, которой уделить внимание?

AЧ: Но которая не находит отражения в моих вопросах? Это лучше у читателей спрашивать. Мне со своей стороны сложно об этом судить. **LXF**

МАШИННОЕ ОБУЧЕНИЕ

Джонни Бидвелл подстрекает вас проявить не одно только праздное любопытство к объединению сил нейронных сетей и обучения с открытым кодом.



О машинном обучении (ML) слышало большинство людей. Если вы временно находились хоть в какой-то близости от Интернета, вам почти наверняка скормили данные, собранные с помощью алгоритмов машинного обучения, а ваш опыт использования будет введен в обучающие модели для развития этих алгоритмов.

В 1958 г., когда была создана одна из первых нейронных сетей для распознавания изображений — Mark 1 Perceptron, мудрецы заявили, что это — «эмбрион электронного компьютера, от которого [ВМФ] ожидает способности ходить, разговаривать, писать, воспроизводиться и осознавать свое существование». С тех пор данная область сильно шагнула вперед, и хотя пока что мы не видели разумных машин (или восстания роботов в стиле Skynet), всё же результаты весьма интересны. ML вносит свой вклад в открытие

лекарств, обнаружение экзопланет и лучшие рекомендации Netflix для всех подряд.

Благодаря доступности огромных наборов данных, таким прорывам в оборудовании, как GPU и TPU, и разработке инструментов (особенно FOSS), которые сильно упростили модели обучения, ML теперь практически повсеместно. Солидный толчок его росту дал промышленно-академический комплекс. Ученые изучают официальные технологии ML с теоретической точки зрения с 1950-х гг. (а некоторые находят зародыши этой технологии аж 200-летней давности). Однако оборудование (точнее, бюджет на создание оборудования), необходимое для выхода на практический уровень, было доступно только компаниям уровня IBM или правительственным лабораториям. Но сейчас, благодаря новым партнерствам, инновациям и крутым облачным технологиям, этот доступ значительно демократизировался.

Обообщенная категория «искусственный интеллект» охватывает множество технологий и легко уведит в область научной фантастики. Где-то за этим названием скрывается дисциплина машинного обучения, которой уделяется большое внимание, причем она умудряется остаться в области научного факта.

ML, в самой свободной интерпретации, относится к любой системе, которая для решения проблемы использует данные, вместо каких-либо предварительно запрограммированных правил или алгоритмов. Это может быть нечто весьма узкое — например, проведение оптимальной кривой через набор точек, или нечто более сложное, как, например, компьютерное зрение или анализ эмоциональности высказываний.

ML ставит традиционное программирование с ног на голову. Вместо ввода данных в алгоритм, который затем выводит ответ, ML берет набор точек данных (пар для входа и выхода) и предлагает нечто вроде связующего их алгоритма, чтобы при вводе новой точки данных можно было правильно классифицировать ее (найти соответствие с выходом).

Не стоит смеяться над примером с кривой — и уж всяко не тогда, когда это работает. Не зная, что, собственно, представляют собой данные, компьютер предложил общее правило, применимое к новым данным для получения значимого результата. Чем больше об этом задумываешься, тем яснее осознаешь, насколько это мощная технология. Представьте себе изумление пифагорейцев, когда их лидер открыл взаимоотношения между катетами и гипотенузой.

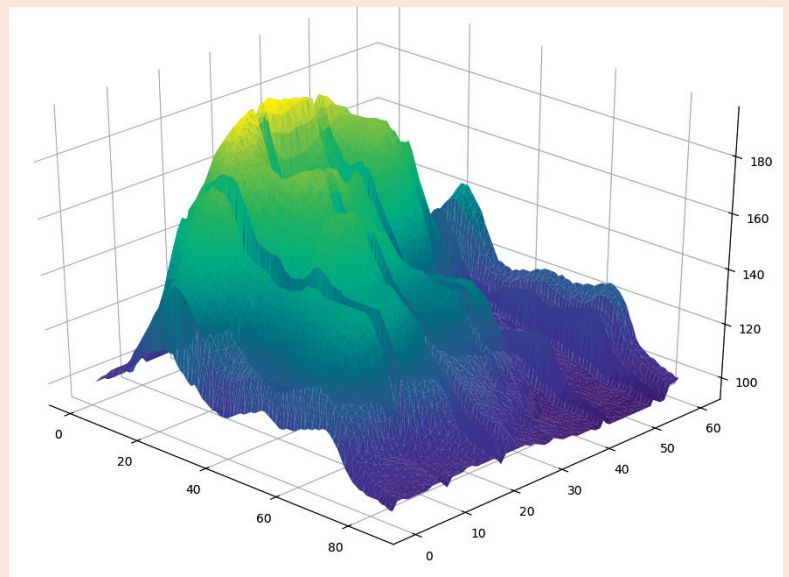
Что стоит за заголовками

Вокруг ML наблюдается сильный ажиотаж. Оно способно решить множество проблем, а в будущем сможет решить еще больше, но вряд ли вам удастся разбогатеть с помощью сляпанного на скорую руку бота, который торгует акциями и ценными бумагами. Точно так же, это никак не инструмент уничтожения человечества. Нам постоянно твердят, что очень скоро вся наша работа будет автоматизирована. Чаще всего в качестве примера приводят рабочих конвейера и водителей, но весьма вероятно, что со временем очень немногие профессии окажутся не затронутыми этой волной автоматизации. Для статей, где фигурируют числа и статистические данные, автоматическая журналистика [Ред.: — О Господи!] уже в ходу. Первое сообщение о землетрясении 2014 г. пришло от системы Quakebot журналиста *LA Times*, Кена Швенке [Ken Schwencke].

Так стоит ли дрожать от страха нашим трудолюбивым писателям в Башнях *Linux Format*? Пока что довольно трудно вывести правило, как найти хорошие, оригинальные идеи для учебника или какие темы осветить, и как объединить всё это в один номер журнала. Кроме того, имеется немало внутренней работы: настройка среды тестирования, поиск разных способов работы и обнаружение неожиданных ошибок. А далее идет собственно само написание статьи, где кто-то черпает из источника своего литературного таланта, одновременно оттачивая собственный уникальный ироничный [Ред.: — И весьма претенциозный] стиль. Пройдет немало времени, пока компьютеры научатся делать то же самое, но область обработки естественного языка развивается очень быстро. А еще очень быстро развивается упадок печатных СМИ, так что бухгалтеры, возможно, доберутся до нас раньше роботов.

Обучение с открытым кодом

По аналогии с большой работой в области теории метода обработки данных (см. наше интервью с Эми Бойл [Amy Boyle] в **LXF207**), избранным оружием FOSS являются Python и NumPy, которые добавляют к вычислительным компьютерным технологиям эффективные структуры. Однако самым именитым в области



машинного обучения является Google TensorFlow, библиотека с открытым кодом, применяемая с Python, C++, OpenCL или любым их сочетанием. Поверх TensorFlow можно использовать библиотеку Keras для нейронных сетей. Популярным был Theano, первым внедривший некоторые технологии, которые ныне являются обычными для инструментов (в частности, прозрачное ускорение GPU), но в прошлом году разработка прекратилась. Наш представитель яйцеголовых, Михалис, опишет некоторые такие инструменты на стр. 74. А мы теперь попытаемся сорвать покров тайны с нейронных сетей, центральной концепции ML...

Поиск минимума от одно- или двумерных функций прост. В нескольких сотнях тысяч измерений это чуть сложнее.

» ОБУЧЕНИЕ МАШИН ИГРАМ

Компьютеры побеждали шахматистов из плоти с того самого момента, когда Deep Blue победил Каспарова в 1990-х. Но в конце 2017 г. мы увидели заголовки, говорящие о том, что движок глубокого обучения от Google, AlphaZero, «сам научился играть в шахматы» за четыре часа, и затем выиграл у Stockfish, лучшего в мире шахматного компьютера (см. www.chess.com/news/view/google-s-alphazero-destroys-stockfish-in-100-game-match).

Это обучение велось на самом передовом оборудовании в мире, что, вероятно, соответствует многим годам обучения на обычном оборудовании. Так что мы не должны вкладывать особый смысл в эти четыре часа. Интересен, однако, стиль игры, показанный AlphaZero. В десяти играх, продемонстрированных зрителям (<https://deepmind.com/research/alphago/alphazero-resources>), он, похоже, не особо переживал по поводу жертв, если была возможность размена фигур, и в некоторых случаях делал ходы, с виду бессмысленные. Это напоминает творческий, непредсказуемый стиль Михаила Таля.

У исследователей ИИ изучает также видеоигры, и компьютеры удивили всех. Например, в докладе (<https://arxiv.org/abs/1802.08842>), опубликованном ранее в этом году, команда из Фрайбурга показала, что происходит, когда в 1982 г. их научили играть в классику Atari, *Q*bert*. Агент (с помощью технологии «эволюционной стратегии») обнаружил в игре ошибку, которую и использовал, чтобы добиться такого результата по очкам, о котором людям и мечтать не приходится.



Вдохновленные миллиардами нейронов в мозге животных и мириадами связей между ними, исследователи создали собственные (хоть и меньшие по размеру) сети, которые своим, искусственным способом могут научиться выполнять определенные задачи. Аналоги биологического мозга очень быстро выходят из строя, однако это не мешает им быть невероятно мощными и полезными.

С одной стороны нашей сети ввод данных представлен уровнем (уровнем входа) искусственных нейронов. Затем этот уровень передает всё на уровень выше, где каждый нейрон соединен со всеми нейронами предыдущего уровня. Каждый нейрон выводит некое значение в зависимости от введенной информации, и это значение передается всем нейронам в следующем уровне. Есть также конечный уровень — уровень выхода, который представляет решение сети.

Чтобы сеть была полезна, необходимо управлять воздействием каждого нейрона на нейроны следующего уровня. Начиная с изначальной конфигурации, мы вводим сетевые данные, и если нам не нравятся выходы, мы регулируем сеть, чтобы улучшить результат. После достаточного обучения, если наша нейронная сеть успешна, мы должны быть в состоянии ввести новые данные,

которые не были частью тренировочного набора, и она должна суметь правильно их категоризировать.

Это довольно сложно понять, но если проиллюстрировать примером, всё становится понятнее. И классический пример — анализ почерка, или, по крайней мере, его поднабора, касающегося распознавания цифр от 0 до 9. Как Михалис упомянул в своем учебнике по кодированию, имеется канонический набор данных (база данных MNIST), который состоит из 70 000 изображений написанных от руки цифр в форме изображений градаций серого 28×28 . Он стал фактическим стандартом для создания «моей первой нейронной сети».

Рассматриваемая проблема

Итак, каждый нейрон в нашем уровне ввода хранит значение градации серого для определенного пикселя. Этот уровень состоит из сравнительно небольшого количества — 784 нейронов, и у каждого — 256 возможных значений (восьмибитные изображения). Оставив пока что в стороне происходящее в промежуточных уровнях; давайте рассмотрим уровень вывода. Нам нужно суметь классифицировать вводимое изображение как одну из десяти цифр, поэтому пусть в нашем уровне вывода будет 10 узлов (мы не будем называть их нейронами), каждый из которых будет измерять «близость» к каждой цифре.

В таком случае в классическом примере неточности списка с 0-индексом, если наша сеть уверена, что мы ввели в нее изображение 1, то значение второго узла уровня вывода должно быть намного выше, чем значения остальных девяти узлов. Поскольку это не точный процесс, ему будет присущ некий шум; остальные результаты не будут в точности равны нулю. Все мы встречались с ситуациями, когда 3 написана так, что похожа на 5, и когда наша сеть сталкивается с подобными случаями, мы рассчитываем, что у узлов 4 и 6 будут более высокие значения, чем у остальных. Если сеть не в состоянии дать ответ, то надо рассчитывать, что у всех узлов вывода значения будут примерно одинаковы.

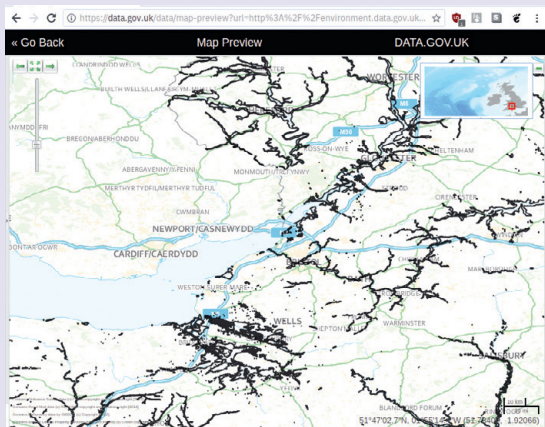
Хорошо. Но нам надо выяснить, что происходит между уровнями ввода и вывода. Что вполне уместно, уровни между ними именуются «скрытыми». Очень соблазнительным представляется придумать, что могут делать последующие уровни, рассмотрев, какие попытки идентификации или описания символа предпринял бы человек. Распознавание контура — удивительно полезный инструмент, поэтому мы можем вообразить где-то внутри нашей сети инструмент, который определяет контуры — круглые, кривые или прямые. Мы также очень любим фигуры (создаваемые определенной конфигурацией контуров), и в цифрах находим ряд их видов: прямые линии, кривые и круги. Поэтому

» БОЛЬШИЕ НАБОРЫ ДАННЫХ

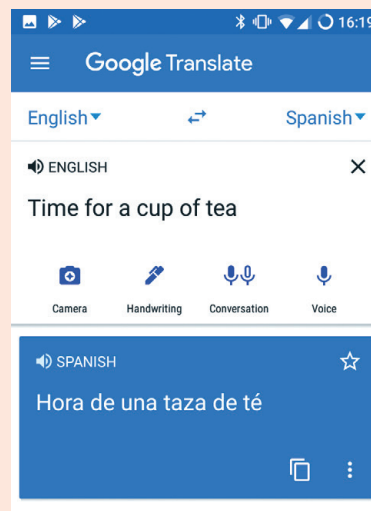
Ключ к машинному обучению — множество данных. Когда-то получение больших данных и возможность их обработки были большой проблемой, но сегодня ставшие нормой широкополосные соединения и великанские жесткие диски делают эту задачу куда проще. Помимо чисел MNIST, есть ряд других полезных баз данных, доступ к которым можно получить бесплатно. Текст Wikipedia объемом 14 ГБ является отличной базой для проектов обработки естественного языка.

Проект DBpedia идет еще дальше, предлагая среду на Java для извлечения более структурированной информации из Wikipedia и относящихся к ней проектов. Они предлагают предварительно извлеченную базу знаний, состоящую из 4,58 миллионов единиц, большинство из которых классифицировано с помощью тщательно продуманных и последовательных объектных моделей. Это позволяет легко обработать отношения между объектами или ограничиться определенной категорией, представляющей интерес, будь то люди, места или растения.

IMDB предлагает ряд наборов данных (www.imdb.com/interfaces) для некоммерческого использования, в том числе большую подборку изображений киноактеров. Изображения самых популярных 100 000 из IMDB и Wikipedia были использованы командой Университета Цюриха для создания самой крупной в мире базы данных лиц. Она была использована для обучения их системы DEX (Deep Expectation), которая оценивает возраст и привлекательность, и оказалась настоящим хитом Интернета. Узнайте больше и скачайте данные на <https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki>.



Правительство Великобритании предлагает множество наборов данных на <https://data.gov.uk>. Здесь — данные по наводнениям.



Google Translate работает в основном на машинном обучении с 2016 г.

мы можем представить себе еще один уровень, который ищет эти характеристики.

Важно помнить, что на самом деле в нейронной сети распознавания цифр ничего подобного не происходит, но всё равно очень удобно представлять всё именно так. В реальных примерах очень трудно предъявлять претензии по поводу того, каких свойств слои располагаются в нейронной сети, и это поднимает ряд философских вопросов. Но, благодаря способу их обучения, на самом деле это не имеет значения.

Избранное алгебраическое оружие

Сведущие в линейной алгебре должны заметить, что каждый уровень узлов может быть представлен чем-то вроде матрицы, где вывод предыдущего ряда представляется своего рода вектором. И это очень просто запрограммировать. Это не совсем та линейная алгебра, которая преподается первокурсникам, поскольку вывод с каждого узла нужно масштабировать и нормализовать. Избранное для этого оружие в математических кругах известно как операторы ReLU (блока линейной ректификации), у которого много общего с работой настоящей нейронной сети: ее вывод нулевой, пока не перейден некий порог активации, и затем он линейно растет.

Раз мы взялись обсуждать терминологию, мы можем также указать, что существуют обобщения матриц, называемые тензорами, и именно отсюда происходит название популярного пакета TensorFlow [поток тензоров]. Взрослое слово для функции, трансформирующей векторные данные — свертка, а тип сети, которую мы только что перекроили по-своему, во взрослом мире называется сверточная нейронная сеть.

Данные MNIST поменены, так что числовое значение каждого изображения легко проверить. Таким образом, легко определить, хорошо ли работает наша сеть: просто посмотрите, насколько отличается уровень вывода от схемы, представляющей число ввода. Это дает нам стоимостную функцию, и минимизируя эту функцию по всем нашим данным обучения, мы делаем нашу сеть максимально полезной. К сожалению, минимизировать эту функцию в общем случае невероятно сложно. Поиск локального минимального значения функции от одной или двух переменных (графиков и плоскостей соответственно) — вычисление довольно элементарное. Но самая маленькая работоспособная нейронная сеть распознавания чисел имеет десятки тысяч переменных (коэффициенты и смещенность матриц, представляющих каждый уровень), и ее обучение включает поиск минимума десятков-тысячемерной поверхности. Трудно даже вообразить, как это выглядит, не говоря уж о том, чтобы это решить.

К счастью, есть цифровые итеративные методы (в частности, метод стохастического градиентного спуска), которые эффективно с этим справляются. Каждая итерация дает нам представление о том, как надо настроить предыдущий уровень, поэтому мы можем прогнать его рекурсивно по сети (это называется обратным распространением) и затем заново протестировать это со всеми данными обучения. Значение нашей стоимостной функции теперь должно быть меньше, но наше путешествие еще не закончено — в целом, весь этот процесс придется повторить многократно для сокращения.

Это требующее солидных вычислений условие на долгие годы лишило нейронные сети популярности. Сегодня вы можете провести числовое распознавание на домашнем компьютере за пару часов (попробуйте пример Keras на **LXFDDVD**), хотя оно использует ненадежные оптимизации (мини-группирование данных обучения). Когда-то это казалось невыполнимым, и многими считалось невозможным. Метод обратного распространения за десятилетия также то входил в моду, то выходил из нее (сейчас он снова в моде). В течение некоторого времени людей заботило



осознание, что метод обратного распространения, по крайней мере, в том виде, как мы его описали, невозможен для нейронов мозга. Однако это не мешает ему оставаться полезным.

Машина – звезда автономии

Новейшие технологии ML сегодня требуют огромного количества вычислений на стадии обучения. Но как только модель пройдет обучение, прогонять через нее данные станет куда проще. Например, следуя модным в машинном обучении словечкам, вы очень быстро угодите в область автомобиль-беспилотников. Однако вождение машины предполагает умение принять жизненно важное решение за считанные секунды. Вряд ли уместно запускать обучение с безалаберным соратником. Сначала мы должны вложить эти тысячи часов и собрать эти тысячи терабайт систем обучения в управляемую среду.

Помимо скорости, модели также переносимы, поэтому вы можете перенести модель классификации изображений Google Inception в TensorFlow или взять голосовое распознавание Snips

Алгоритм машинного обучения, возможно, поднимет вас на смех за то, что вы не продали свои BTC еще в декабре, но что он понимает?

ВЫБЕРИТЕ ЛУЧШИЕ МОМЕНТЫ ML «Вождение машины предполагает умение принять жизненно важное решение за считанные секунды. Вряд ли уместно запускать обучение с безалаберным соратником.»

(см. <http://bit.ly/snips-weather>) и запустить его на Raspberry Pi (возможно, в работе). Вы даже можете запустить на нем простые обучающие примеры, но это потребует времени. Портинг TensorFlow на Pi (2 и выше) было феноменальным достижением: он использует тяжеловесную систему сборки Bazel, так что его портинг было первым шагом. Затем надо убрать из кода TensorFlow все вызовы 64-битных библиотек, а разные зависимости требуют применения Band-Aids. Даже на Pi 3 компиляция занимает несколько часов, но эта работа была сделана в 2016 г., до выхода Pi 3. Так что снимите шляпу перед Сэмом Эйбрахамсом в благодарность за его терпение.

Если только у вас нет особых требований, нет нужды повторять его усилия. Модуль Python TensorFlow доступен через *pip*, так что его можно установить в мгновение ока. **LXF**

DTrace для Linux и управление с cgroups



НАШ
ЭКСПЕРТ

Д-р Валентин Синецын — бывший разработчик KDE с 0,16 практикантами Google в год, так что у него достаточно времени на то, чтобы создавать облака Linux и писать слова.

Валентин Синецын недоумевает: почему Oracle не нужна открытость насчет внесения изменений в бывшие технологии Sun, хороших или плохих?

3 аглядывать в работающие производственные системы — всё равно что менять шины на движущемся автомобиле: возможно, но экзотично. В **LXF231** мы рассмотрели *eBPF*, позволяющий проводить сессии проверки и профилирования на системном уровне, безопасно и быстро. Мы также пояснили вам, что *eBPF* концептуально является преемником *DTrace*, технологии Solaris, появившейся в начале 2000-х. В том же выпуске мы рассказали, что Oracle потихоньку закрывает Solaris и что не стоит больше ждать оттуда новостей. И вот это последнее было не вполне верно...

Когда Sun Microsystems сделали открытым код большей части Solaris, включая *ZFS* и *DTrace*, частично это объяснялось борьбой с растущей популярностью их соперника Linux с открытым кодом. Поэтому Sun выбрали свободную, но несовместимую

с GPLv2 лицензию: CDDL. Нет единой точки зрения на то, может ли *ZFS* или *DTrace* быть частью ядра Linux или нет, поэтому причины их невключения были скорее юридическими, чем техническими. На самом деле, FreeBSD уже долгое время включает их обе.

Теперь Oracle, приобретя права на технологии Sun, молчком делает еще один смелый шаг. Они повторно лицензируют код ядра *DTrace*, соответствующих библиотек и инструментов под GPLv2 и UPL. Настоящее изменение произошло в августе 2017 г. — но произошло настолько тихо, что оставалось незамеченным до февраля 2018 г. UPL означает Universal Permissive License (она совместима с GPL). Есть вероятность, что Oracle работает над интеграцией *DTrace* в основное ядро. Ранее *DTrace* стал доступен для Linux в качестве части ядра Oracle Unbreakable Linux благодаря специальному исключению CDDL, сделанному Oracle для себя.

Конечно, действия Oracle радуют. Но, цитируя Бернарда Шоу: «[Нобелевская премия] — это спасательный круг, брошенный тому, кто уже благополучно доплыл до берега». То же самое относится к *DTrace* под GPL. Десять лет назад он мог бы дать толчок Linux, но сейчас он скорее просто помогает избавиться от балласта. Пока Oracle стерег драгоценности своей короны, мы приобрели собственные сокровища. И теперь даже использование такого высокоуровневого языка D, как *DTrace*, уже не кажется убийной функцией.

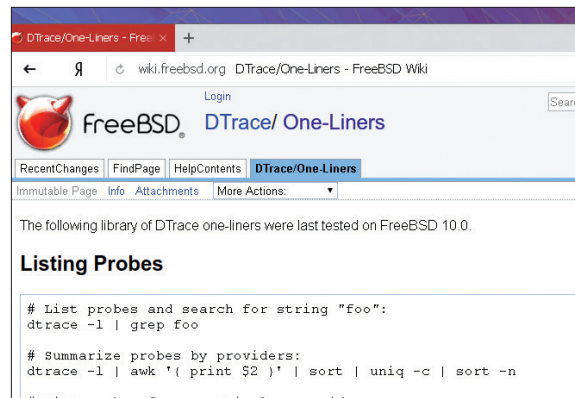
» КАК ВСЁ РАБОТАЕТ

Представьте себе, что все ваши знания заключены в сферу. Чем их больше, тем больше радиус сферы. Поверхность этой сферы касается предметов, о которых вы осведомлены, но которые пока не изучили. Когда вы учитесь, сфера становится больше, но площадь ее поверхности растет еще быстрее. И чем больше вы знаете, тем больше вам приходится изучать. Именно по этой причине я прошел курс информатики в одном из онлайн-университетов.

Я немало узнал, и мне понравился курс, хотя и не весь. Курс дал мне немало идей и представлений, но слабовато обосновывал, почему я должен делать что-то так, а не иначе.

Подобное обучение в стиле кулинарной книги неплохо, но ему не хватает гибкости. Это как учить заклинания. Конечно, любая достаточно продвинутая технология неотличима от магии, и всё же она остается технологией. Впервые увидев *Vim*, вы не знаете, как из него выйти. Попытавшись понять логику его горячих клавиш, вы начинаете редактировать тексты быстрее, чем раньше. Возможно, вы могли бы добиться таких же результатов, просто выучив горячие клавиши, но для меня лично это не работало, сколько бы я ни старался.

Вот почему мне кажется важным понять, как всё работает. В этом месяце мы рассмотрим, как Linux реализует группы управления. Возможно, вам вообще не придется сталкиваться с интерфейсами, о которых мы расскажем, но, надо надеяться, это поможет вам лучше понять границы *Docker* и других инструментов. Что хорошо знать само по себе.



FreeBSD использовал массу прелестей Solaris, которых официально нет в Linux, по крайней мере, пока. Одним аргументом меньше в будущих ожесточенных спорах.

cgroups: “с” означает “control”

Иногда ресурсы бывают ограничены, так как насчет укрепления этих границ? Вот вам способ укротить аппетит процессов Linux...

Те из вас, кто внимательно следит за технологическими тенденциями, знают, что крутой новинкой являются контейнеры. Вы используете их для компиляции, доставки и развертывания частей кода, обычно в виде микросервисов. Не обращайте внимания на немного ироничный тон, потому что контейнеризация — действительно удобный подход. Но при этом мы всё равно чувствуем себя одиноко, называя это модным маркетинговым словом. Почему? Потому что в Linux нет такой вещи, как контейнер.

По сути, контейнеры можно свести к двум концепциям. Одна — область имен, и именно так они добавляются изоляции. Например, область имен процесса обеспечивает, что pid 1 в вашем контейнере и pid 1 в ОС вашего хоста — две разные вещи. Вторая — группа контроля, или “cgroups”. Это механизм, обеспечивающий разумное потребление ресурсов, что бы вы ни считали разумным. Фактически, у cgroups много областей использования помимо контейнеров.

В Рецептах этого месяца мы выясним, как они работают в *systemd*, *Docker* и т. п.

Тегируйте их

Cgroups также двойственны. Они являются иерархическими областями имен ярлыков, которые вы можете присваивать процессам (на самом деле всё по-другому). И они также являются контроллерами, укрепляющими ограничения в каждой иерархии ярлыков. Т.е. дочерние ярлыки содержатся внутри границ, установленных их родителями. Таким образом можно установить ограничения для каждой сессии пользователя и для каждого процесса в каждой сессии.

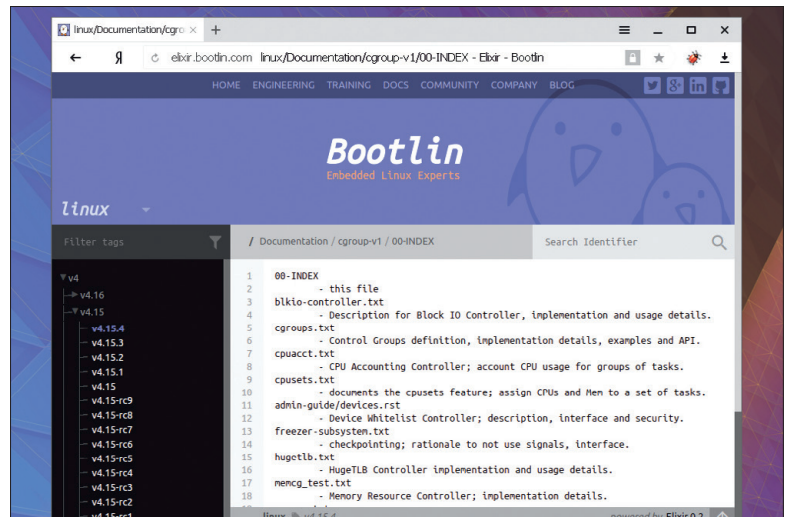
Иерархические структуры естественно проецируются на деревья файловых систем, и именно этим путем идут cgroups. Ядро предоставляет вам файловую систему “cgroup”, которую вы можете смонтировать куда угодно, хотя на сегодняшний день обычно выбирается `/sys/fs/cgroup` (сама по себе tmpfs). С *systemd* она монтируется по умолчанию. Если такого не произошло, сделайте это сами, как с любой другой файловой системой:

```
$ sudo mount -t cgroup -o cpu cgroup /some/where
```

Здесь “cpu” — имя контроллера. Вы также можете употребить “all”, для монтирования всех имеющихся контроллеров. Если эта команда говорит “cgroup already mounted”, это означает, что *systemd* или что-то еще уже смонтировали контроллер за вас. Если контроллеры A, B, и C смонтированы, можно спокойно размонтировать все три, но не единственный A (или B, или C). Чтобы размонтировать контроллер, он должен быть «занят [busy]» — т.е. к нему должны быть приписаны процессы. Короче говоря: если у вас уже есть `/sys/fs/cgroup`, используйте для нашего урока его.

Чтобы всё немного усложнить, есть две версии cgroups: v1 и v2. Map-страница `cgroups(7)` объясняет разницу, но на данный момент достаточно знать, что в cgroups v2 все контроллеры монтируются в одной унифицированной иерархии. *Systemd* делает это в `/sys/fs/cgroup/unified`. Вы не можете монтировать один контроллер в обеих иерархиях v1 и v2. Более того, cgroups v2 по-разному обращаются с потоками и предлагают лучший механизм событий, но это уже другая история...

Смонтировав cgroups в `/sys/fs/cgroup` или где-то еще, можете приступить к созданию ярлыков (иерархий) и присваиванием им процессов:



```
$ sudo mkdir /sys/fs/cgroup/cpu/my-group
$ echo $$ | sudo tee /sys/fs/cgroup/cpu/my-group/cgroup.procs
14130
```

Последняя строка — то, как вы, не являясь root, пишете в файл, принадлежащий root. Вот как: вы переместили текущий процесс (вероятно, оболочку) в `/my-group`. См. следующее:

```
$ cat /proc/self/cgroup
```

```
...
```

```
4:cpu,cpuacct:/my-group
```

4 — ID номер непрозрачной иерархии. `cpu` и `cpuacct` — имена контроллеров: они идут бок о бок, когда *systemd* монтирует их на моей системе.

Вам не надо быть хакером ядра, чтобы читать документацию ядра Linux; здесь — официальная ссылка на cgroups.

» CGROUPS ДЛЯ ПРОДВИНУТЫХ

Контроллеры “cpu” и “memory” упоминаются в каждом уважающем себя руководстве по cgroups, но Linux может предложить куда больше.

Популярен контроллер “blkio”, он ограничивает количество запросов на блочный ввод/вывод, разрешенное иерархии процесса, и используется, чтобы большие программы, считывающие (или записывающие) множество данных, не обделяли ресурсами маленькие программы. Существует две политики. Первая называется «разделение пропорционально весам», и она похожа на контроллер “cpu”: вы указываете вес, и “blkio” контроллер делит ресурсы соответственно. Другая — «политика верхнего предела», и она реализует троттлинг [регулирование количества запросов], если процессы в cgroup превышают установленный уровень байтов в секунду для блочного устройства.

Еще один ресурс — идентификаторы процессов (PIDs). Если все они определены, то вы не сможете создать процесс, независимо от того, сколько у вас свободных тактов CPU и памяти. Контроллер “pids” позволяет установить максимальное количество процессов в иерархии, и системные вызовы `fork(2)` или `clone(2)` не могут добавить больше. Однако это не применяется к процессам, которые вы добавили, так что наличие 12 процессов в группе с ограничением до 10 — это нормально. Они смогут запускать новые программы, но любая попытка `fork()` выдаст -EAGAIN. Если в иерархии более одной группы, применяется самое строгое ограничение.

Cgroups наследуются в ответвлениях, так что если вы выдаете команду в оболочке, в конечном итоге она окажется в той же самой cgroup:

```
$ tail -n 2 /sys/fs/cgroup/cpu/my-group/cgroup.procs
14130
14439
```

14130 — PID оболочки, и когда мы запускаем “tail” для чтения процессов в cgroup оболочки, он видит, что его собственный PID (14439) добавился автоматически.

Установите защиту

Cgroups обеспечивают контроллер на каждый тип ресурсов, например, CPU или память. Иерархии процесса — это директории. Вы пишете в них, чтобы укрепить ограничения ресурсов, и читаете их для определения текущих настроек или статистики. Есть десятки контроллеров cgroup v1 (см. врезку *справа*), но рассказать о них всех будет утомительно. Вместо этого мы рассмотрим ряд наиболее простых. Обо всех остальных вы сможете прочитать в документации ядра Linux.

Давайте начнем с уже упомянутого “cpu” контроллера. Как мы видели, *systemd* монтирует его вместе с “cruacct”. Контроллер “cpu” управляет так называемыми “CPU shares [распределением ресурсов CPU]” через файл *cpu.shares*. Обычно каждому процессу отводится 1024 доли ресурсов, и вы увеличиваете или уменьшаете это число, чтобы отвести ему больше или меньше времени CPU. Однако это работает только когда CPU занят. На простаивающем чипе время процесса не ограничено: нет конкуренции, нет ограничений. Вы можете увидеть, был ли троттлинг процессов в группе и на какой срок, в файле *cpu.stat*:

```
$ cat /sys/fs/cgroup/cpu/my-group/cpu.stat
nr_periods 0
nr_throttled 0
throttled_time 0
```

nr_periods — сколько раз нужен планировщик для укрепления ограничений ресурсов. *nr_throttled* — сколько раз для этого были ограничены ресурсы в группе. Это разные числа, потому что планировщик мог решить подвергнуть троттлингу что-то еще. И, наконец, *throttled_time* — продолжительность троттлинга группы. Ноутбук, который мы в данный момент используем как печатную машинку, не подвергается большой нагрузке (мы не так уж быстро печатаем), поэтому во всех этих полях у него ноль.

“cruacct” — пример контроллера, который ничего не защищает. Вместо этого он выполняет учет CPU по иерархии: статистика собирается сверху вниз. И если у вас есть, скажем, */services*,

/services/NetworkManager, */services/cups* и так далее, вы мгновенно получите статистику по каждому сервису и для всех сервисов (в противоположность пользовательским программам) вместе взятых. Иными словами, “cpu” устанавливает ограничения ресурсов, а “cruacct” сообщает реальное потребление; неудивительно, что они идут рука об руку.

На каждом уровне управляемой *cruacct* иерархии имеется файл *cpuacct.usage*, содержащий количество наносекунд процессов в этой группе и ее дочерних, потраченных на CPU, так что */sys/fs/cgroup/cpuacct.usage* дает время CPU, отведенное всем потокам в системе. Другие файлы *cpuacct.** собирают статистику по каждому процессору или пользователю/системе: опять же, все подробности в документации ядра.

Ограничение памяти

Говорят, в наши дни нет такого ресурса, как память. Тем не менее, есть контроллер “memory”, ограничивающий ее для вас. Linux всё же упорно привержен экономии памяти, так что проблема не в том, чтобы запросить много памяти: проблема возникает, когда вы начинаете ее использовать.

Есть несколько рычагов, на которые можно надавить, чтобы ограничить обычную память процесса, память ядра и область подкачки. Файл под названием *memory.limit_in_bytes* настраивает ограничения памяти процесса, *memory.memsw.limit_in_bytes* добавляет область подкачки, а *memory.kmem.limit_in_bytes* ограничивает память ядра. Последний обычно включает стеки режима ядра, объекты, распределенные ядром от лица процессов (например, открытие файлов), и сокеты. Для сокетов возможен более тонкий и тщательный контроль, так что вы можете ограничить память, используемую для буферов TCP.

Хотя ограничения выражаются в байтах, при их настройке можно использовать другие единицы, например, МБ или ГБ:

```
$ sudo mkdir /sys/fs/cgroup/memory/my-group
$ echo 4G | sudo tee /sys/fs/cgroup/memory/my-group
$ cat /sys/fs/cgroup/memory/my-group
4294967296
```

Обратите внимание, что нельзя установить ограничения на cgroup памяти root.

Что произойдет, если процесс выйдет за пределы cgroup? Основная процедура такая же, как и для обычного, не ограниченного процесса. Сначала ядро попытается затребовать еще памяти и подкачать ее в процесс. Если это не поможет, вызывается OOM killer, чтобы остановить самый наглый процесс в cgroup (надо надеяться).

Можно также отменить ограничения памяти. Для этого просто напишите -1 в соответствующем файле. При чтении значения будет выведено 9223372036854771712. Это ровно половина от 2⁶⁴ минус одна страница. Некоторые могут сказать, что это максимальный адрес области пользователя, который когда-либо был у 64-битного приложения (однако не на современных процессорах). Остальные могут просто считать его бесконечностью.

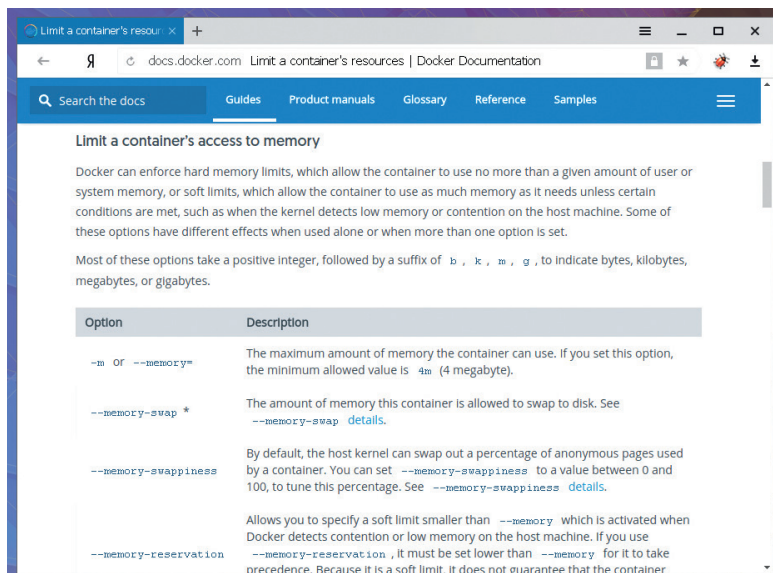
Пример из реальной жизни

Теперь у вас есть некоторое представление о том, как управлять cgroups вручную. Это поможет создать полную картину (что всегда хорошо), но практически не несет пользы: вы не управляете cgroups таким образом. Вместо этого вы используете высокоуровневое управление, как то, которое имеется в *systemd* или *Docker*.

Давайте начнем с первого. Если у ядра хоста включен контроллер “cpu” (ныне это обычно для большинства стандартных ядер), то *systemd* автоматически добавит cgroup для каждого запускаемого сервиса:

```
$ ls -l /sys/fs/cgroup/cpu/system.slice
```

Если вам кажется, что ограничения памяти в Docker подозрительно похожи на то, что делает контроллер “memory” cgroup, то вы абсолютно правы.




```
networking.service
systemd-journald.service
...
```

В реальном выводе более 60 строк в системе Kubuntu 17.10. Это уже дает хороший побочный эффект. Помните, что все cgroups получают одинаковую долю ресурсов CPU по умолчанию — 1024. Это означает, что все сервисы получают одинаковое количество мощности, которую Леннарт [Lennart] считает разумной долей по умолчанию. Если вы не согласны, просто задайте CPUShares другую величину в файле настройки соответствующего сервиса. Ограничения памяти устанавливаются так же: systemd.resourcecontrol(5) содержит все подробности.

На самом деле, ни один из сервисов не переопределяет ограничения в наших системах. Поэтому давайте посмотрим, как Docker управляет за нас ресурсами. У нас установлена версия 17.09.1-се. Потом мы случайным манером выбрали образ из хранящихся локально и создали контейнер, который получил ID 4d1f57c49ef5, вот так:

```
$ docker run --rm=true --cpu-shares=512 --memory=4G
...
```

Заглянув в `/sys/fs/cgroup`, мы увидим, что Docker для каждого контроллера создал иерархию `/docker/<container ID>`:

```
$ find /sys/fs/cgroup -name 4d1f57c49ef5*
...
/sys/fs/cgroup/memory/docker/4d1f57c49ef5...
/sys/fs/cgroup/cpu,cpuacct/docker/4d1f5...
...
```

Проверка соответствующих файлов контроллера показывает, что Docker применит ограничения, которые мы установили в командной строке:

```
$ cat /sys/fs/cgroup/cpu,cpuacct/docker/4d1f5.../cpu.shares
512
$ grep "" /sys/fs/cgroup/memory/docker/4d1f57c49ef5.../*
limit_in_bytes
/sys/fs/cgroup/memory/docker/4d1f57c49ef5.../memory.kmem.
limit_in_bytes:9223372036854771712
/sys/fs/cgroup/memory/docker/4d1f57c49ef5.../memory.kmem.tcp.
limit_in_bytes:9223372036854771712
/sys/fs/cgroup/memory/docker/4d1f57c49ef5.../
```

» НЕМНОГО ОБЛАСТИ ИМЕН

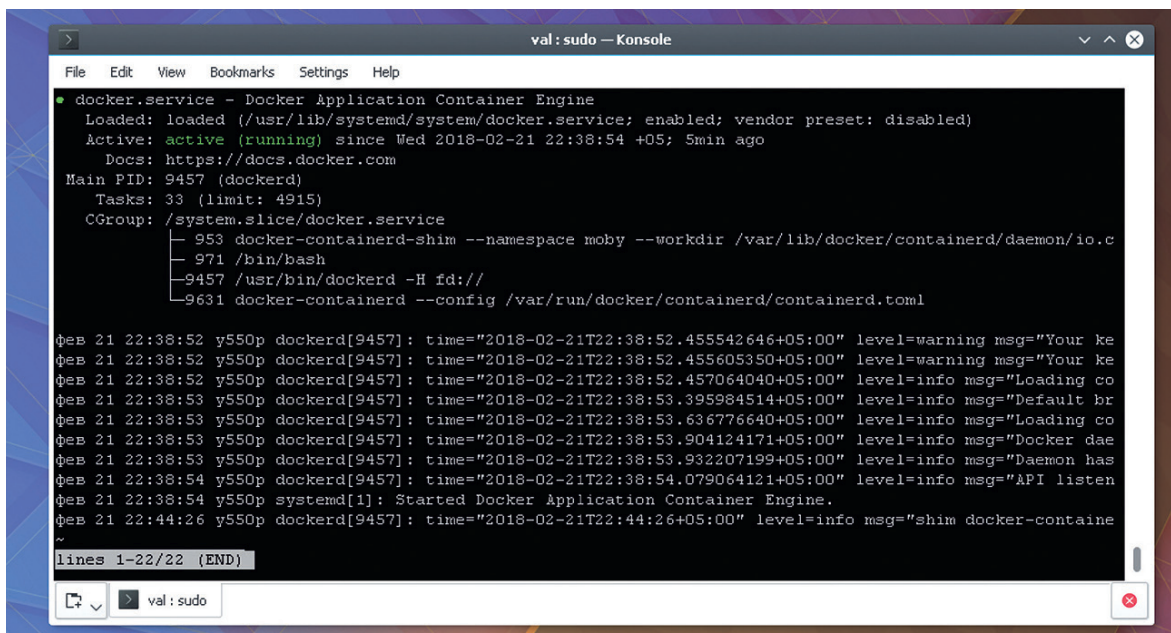
Хотя Рецепты этого месяца посвящены cgroups, мы сочли, что будет несправедливо упомянуть Docker, проигнорировав область имен. Итак, краткое резюме. Как и cgroups, область имен в Linux имеет много разновидностей. Многие из нас слышали о сетевых областях имен. Они являют собой популярный способ получить отдельные сетевые стеки на одном компьютере. Сюда входят стеки IPv4 и IPv6, чтобы вы могли распределить несколько адресов `192.168.0.1`, имеющих разное значение для разных процессов. Обычно вы создаете новые сетевые области имен с помощью `ip netns create` и запускаете команды в области имен по `ip netns exec`. Обычный подход — создать пару виртуальной Ethernet (veth) и отдать один конец области имен, оставив другой хосту (изначальная сетевая область имен).

Существуют также и другие области имен. С помощью области имен монтирования вы можете создать отдельные деревья файловой системы root, возможно, с несколькими директориями с распределенным доступом. Область имен пользователя — это то, что отличает пользователя root в контейнере от пользователя root в системе, что крайне важно для безопасности. С помощью области имен UTS вы можете изолировать внесенные вами изменения в имя хоста. Область имен IPC защищает очереди сообщений, семафоры и т.п., и мы уже упоминали область имен PID в основной части. И последнее — но далеко не менее значимое — это область имен cgroup, которая создает новую виртуальную cgroup root для процесса. Тогда подробности управления хоста не открываются непривилегированному контейнеру.

```
memory.limit_in_bytes:4294967296
```

Обратите внимание, что ограничена только память процесса. В нашей системе нет поддержки ограничения подкачки, поэтому она не ограничена, и мы также решили не ограничивать память ядра.

Теперь вы понимаете, что такое cgroups и чем они могут быть полезны — даже для тех из нас, кто не следует трендам индустрии и по-прежнему применяет пакеты. Если вы хотите, чтобы ваш сервис потреблял не больше данного CPU, или решили задать количество памяти, запустить определенные библиотеки или обеспечить легковесность ввода/вывода диска, то cgroups должны стать вашим первым помощником. **LXF**



Команда `systemctl status` также показывает информацию о cgroups для сервиса. Здесь вы видите четыре процесса, относящиеся к демону Docker.

» ПОДПИШИТЕСЬ НА ЖУРНАЛ LINUX FORMAT на www.linuxformat.ru/subscribe/

Подпишись на Linux Format



Печатная версия

Годовая подписка
4500 ₽*

Полугодовая подписка — 2430 ₽*

БОНУС

Подписка на PDF-версию журнала

ДОСТАВКА

200 пунктов самовывоза в 50 городах России

Журнал
Linux Format
издается
в России
с 2005 года



Электронная версия (PDF)

Годовая подписка
2760 ₽

Полугодовая подписка — 1485 ₽

БОНУС

Образ DVD-приложения к каждому номеру

БОНУС

Получение каждого номера в день выхода журнала

* Цена не включает стоимость доставки.

ПОДРОБНОСТИ НА САЙТЕ
WWW.LINUXCENTER.RU/LINUXFORMAT



Учебники



Евгений Балдин
Подтвердивший
свою квалификацию
физик.

» БОЛЕЗНЬ ГРЯЗНЫХ РУК

Но нельзя бороться за чистое
грязными руками.
Сергей Олегович Дорофеев

Я благодарен Canonical за то, что они сделали и продолжают делать для сообщества. Я использую дистрибутив Ubuntu (сейчас 16.04, следующий LTS попробую через полгода) на постоянной основе и дома, и на работе.

Эта история о том, как начинающий пользователь GNU/Linux установил Ubuntu и воспользовался ее активно продвигаемым Ubuntu Snap Store для установки простой головоломки 2048. После этого он обнаружил, что процессор его компьютера используется на 100% постоянно. Вместо того, чтобы махнуть на этот факт рукой, пользователь зарегистрировался на linux.org.ru и под ником AlexSav попытался выведать у местного дикого сообщества: «А в чем, собственно говоря, может быть проблема?» Внезапно нашелся терпеливый и компетентный пользователь slvrn, который точными вопросами диагностировал проблему, а именно то, что вместе с головоломкой 2048 на компьютер пролез майнер. После этого пользователем Krop4ek был сделан пост “Caution! There are malware Snaps in Ubuntu Snaps Store” на Reddit, и огласка пошла по нарастающей.

Следует отметить, что это показательный пример, когда сообщество приносит помощь всем с помощью тысяч глаз, следящих за необычным. Все пакеты от Nicolas Tomb (ник загрузившего заразу) были прибиты, но проблема с помойкой в Ubuntu Snap Store остается. Первыми ломают не популярных. Первыми ломают безалаберных. Ubuntu Snap Store в том виде, как он навязывается сейчас, должен быть уничтожен.
E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...

WATSON

Следить за временем 56
Испугавшись, что он тратит слишком много времени на обед, **Шашанк Шарма** ухватился за утилиту *Watson*, и теперь любителю доскональными отчетами о каждой затраченной минуте.

СЕТИ

Компилировать удаленно 58
Матс-Таге Аксельссон пожалел перегруженный компьютер и раскидал его задачи по другим машинам. Компиляция — неплохой вариант для распараллеливания процесса.

NEXTCLOUD

Интегрироваться с облаком 62
Пользуясь чужими облачными сервисами, вы утрачиваете контроль над своими данными. **Владимир Диаконов** запустил свой сервер документов и подключил к нему офисный пакет.

ОБЛАЧНОЕ ХРАНЕНИЕ

Шифровать облака 66
А если всё-таки решиться отдать данные на сторону, то уж лучше их зашифровать. **Брайан Морк** разъясняет, как бесплатно завести конфиденциальную облачную резервную копию.

NODE-RED

Автоматизировать быт 70
Тим Армстронг общается со своими бытовыми приборами через сайт, отдает им приказания, снимает с них потоки данных и эти данные визуализирует. У Вещей собственный Интернет!

АКАДЕМИЯ КОДИНГА

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

PYTHON 3

Учиться и учиться 74
С помощью машинного обучения **Михалис Цукалос** обнаруживает скрытые в данных шаблоны и информацию, которые в противном случае трудно было бы распознать. А вы учите матстатистику...

WORDPRESS

Чудеса WordPress 78
Благодаря **Кенту Ельчуку** и шаблонам сайтов вы сможете создать свое собственное меню со ссылками на отдельные страницы или категории блога. И сайт будет принадлежать исключительно вам.

Следим за временем

Отслеживание времени, затрачиваемого на проект — Геркулесов труд. К счастью, **Шашанк Шарма** знает быстрое и простое решение.



НАШ ЭКСПЕРТ

Шашанк Шарма — выступающий в суде адвокат из Дели и ярый фанат Arch Linux. Он всегда в поиске предметов для коллекции технаря.

Если вы не очень-очень собранны, то вряд ли сможете точно описать затраты времени на любую задачу, выполняемую за определенный период. Аккуратное отслеживание времени, затрачиваемого на выполнение задания, становится приоритетным, если выставлять счета клиентам планируется на почасовой основе.

Приложение для отслеживания времени поможет вам понять, как тратятся рабочие часы. Эти данные поспособствуют нахождению областей, в которых необходимо повысить производительность. Возможно, вы поймете, что вы чересчур долго обедаете или у вас уходит слишком много времени на исследование, и некогда отвечать на электронные письма (хотя, конечно, мы все знаем, что некоторые электронные письма можно безопасно игнорировать [Ред.: — А вот с этим поосторожнее] или, еще лучше, сразу удалять.

Watson — утилита командной строки, написанная на Python и выпущенная под лицензией MIT. Она позволяет легко

```

)
Options:
-c, --current / -C, --no-current (Don't) include currently running frame in report.
-f, --from DATE The date from when the report should start. Defaults to seven days ago.
-t, --to DATE The date at which the report should stop (inclusive). Defaults to tomorrow.
-y, --year Reports activity for the current year.
-m, --month Reports activity for the current month.
-w, --week Reports activity for the current week.
-d, --day Reports activity for the current day.
-p, --project TEXT Reports activity only for the given project. You can add other projects by using this option several times.
-T, --tag TFXT Reports activity only for frames containing the given tag. You can add several tags by using this option multiple times.
-j, --json Format the report in JSON instead of plain text.
--help Show this message and exit.
linuxlala@playground-budgie:~$
  
```

У некоторых команд много опций, и *Watson* предоставит их все с помощью параметра `--help: watson report --help`.

»» ОПЦИИ НАСТРОЙКИ

Многозадачность — злейший враг отслеживания времени. Это связано с тем, что инструменты типа *Watson* не в состоянии одновременно записывать действия в разных проектах. Конечно, с одной командой `watson start` можно использовать несколько тегов, но, к сожалению, нельзя записывать два проекта одновременно.

Всегда следует помнить об остановке расчета времени проекта командой `watson stop`. Или настройте *Watson* на остановку записи текущего проекта, когда дается новая команда `watson start`.

Есть два способа это сделать. Во-первых, можно запустить команду `watson config options.stop_on_start TRUE`. Во-вторых, можно запустить команду `watson config -e`, которая открывает файл настройки в заданном редакторе, и добавить в этот файл следующие строки:

```
[options]
stop_on_start = TRUE
```

При изменении опции настройки из терминала формат команды получается такой: `options.имя_опции ЗНАЧЕНИЕ`. Значение может быть TRUE, FALSE, 1 или 0. Те же самые параметры, которые определены в файле настройки по команде `watson config -e`, должны быть размещены под заголовком `[options]`, как показано в блоке кода.

Руководство пользователя на официальном сайте *Watson* описывает несколько других настраиваемых параметров, таких как формат времени и даты, и должен ли инструмент генерировать отчет для текущего проекта или для всех проектов, в которых вы участвуете в данный момент.

отслеживать время, потраченное на проекты и задания, и умеет создавать отчеты о собранных данных и показывать, как вы проводили рабочие часы в любой день, неделю, месяц или даже год.

Элементарно, дорогой читатель

Проект размещен на GitHub, можно легко клонировать репозиторий и установить *Watson*. Для большинства дистрибутивов самое простое установить опцию использования диспетчера пакетов *pip*. Если вы еще не установили его, можно запустить команду `sudo apt install pip` или `su -c 'dnf install pip'` на своих машинах Ubuntu и Fedora. Большинство других дистрибутивов также предоставляют *pip* в своих репозиториях, и можно беспрепятственно его установить. Затем можно установить *Watson* командой `sudo pip install td-watson`.

Инструмент автоматически извлечется и установится в каталоге `~/local/bin`. Поскольку этот каталог не описан в `$PATH`, использовать *Watson* пока не получится. Запустите команду `echo 'export PATH="$HOME/local/bin:$PATH"' >> ~/.bashrc`, которая добавляет этот путь в ваш файл `.bashrc`. Потом надо перезагрузить файл `.bashrc` командой `source .bashrc`.

Одно из лучших свойств *Bash* — автодополнение по клавише Tab, и вы можете настроить под него *Watson*, загрузив файл `watson.completion` из репозитория GitHub и поместив его в каталоге `etc/bash.completion.d/`. Перезапустив терминал оболочки, вы сможете использовать автодополнение с *Watson*. Для начала введите `watson [tab] [tab]` в терминале, чтобы получить список всех доступных команд.

Для каждой задачи, которую вы хотите записать с помощью *Watson*, надо указать название проекта, и в каждом проекте допускается иметь несколько разных задач. Например, проект с именем *Leisure* [Досуг] можно применять для отслеживания времени, ушедшего на внерабочие мероприятия. К задачам этого проекта относятся чтение [reading] и видеоигры, и в рамках проекта они помечаются тегами. В разных проектах можно использовать одинаковые теги.

При каждом запуске команды `watson start` проект записывает все соответствующие данные, такие как имя проекта, соответствующие теги и время начала, и сохраняет его в такой штуке, которая в проекте называется фреймом [frame, *англ.* рамка].

Когда вы командуете `watson stop`, проект прекращает запись и выводит идентификатор [id] фрейма последнего записанного сеанса. Можно получить список всех записанных фреймов, запустив команду `watson frames`. Само по себе имя фрейма не несет никакой информации о содержащихся в нем данных, таких как имя проекта или время начала и завершения.

Начало работы

Прежде чем вы сможете начать генерировать отчеты о том, как проводите рабочие часы, надо определить проекты и задачи, которые вы хотите записать с помощью *Watson*. После этого начинайте запись сеанса с помощью `watson start` имя_проекта +теги. Каждая команда запуска поддерживает использование нескольких тегов:

```
$ watson start Leisure +reading
```

```
Starting project Leisure [reading] at 19:53
```

Первая команда велит *Watson* записать проект с именем *Leisure*. Теги не являются обязательными, но мы приставили здесь теги, чтобы определить тип действия на досуге. *Watson* предоставит имя фрейма для самого последнего записанного действия при запуске команды `stop`:

```
$ watson stop
```

```
Stopping project Leisure [reading], started 12 minutes ago. (id: c382b06)
```

Для каждой новой команды запуска *Watson* создает новый фрейм, даже если вы перезапускаете уже существующий проект. В команде `start` можно использовать несколько тегов или использовать разные теги во фреймах одного и того же проекта, например, используя теги `+Coffee` и `+Lunch` в разное время для записи перерывов [Break], которые вы делаете в течение дня:

```
$ watson start Break +lunch
```

```
Starting project Break [lunch] at 13:15
```

```
$ watson stop
```

```
Stopping project Break [lunch], started 30 minutes ago.(id: 0d8583d)
```

```
$ watson start Break +“вечерний чай”
```

```
Starting project Break [evening tea] at 19:25
```

Как видите, проект *Break* мы использовали два раза с разными тегами. Можно запускать команды `watson frames` и `watson projects`, чтобы отобразить список, соответственно, всех фреймов и всех проектов. Не беспокойтесь о нескольких фреймах, созданных для одного и того же проекта и с одинаковыми тегами. При создании отчетов *Watson* автоматически суммирует количество времени, затраченного на любой проект или задачу, по всем таким фреймам.

Отчеты о дежурстве

Можно использовать *Watson* для создания ежедневных, еженедельных или ежемесячных отчетов. Если вам нужны отчеты только для определенного проекта или за определенный период времени, например, 50 дней, можно сделать и это. Команда `watson report` отобразит все детали проекта:

```
$ watson report
```

```
Thu 15 February 2018 -> Thu 22 February 2018
```

```
Break - 2h 23m 34s
```

```
[вечерний чай 2h 23m 34s]
```

```
Doctor's-Visit - 2h 21m 11s
```

```
LXF235 - 30h 04m 08s
```

```
[Сравнение 17h 01m 37s]
```

```
[Учебник 13h 02m 31s]
```

```
Leisure - 1h 04m 11s
```

```
[comics 02m 32s]
```

```
[reading 1h 04m 11s]
```

```
Lunch - 1h 20m 48s
```

```
[break 1h 20m 48s]
```

```
Total: 36h 22m 38s
```

О нет! Если рассмотреть приведенные выше данные, они разоблачат наше пристрастие к вечернему чаю! По умолчанию *Watson* предоставляет данные за последние 7 дней. Здесь был отмечен период с 15 по 22 февраля. Как можно видеть, обед показан как всего лишь 1 час 20 минут; это означает, что за неделю мы не раз забывали записать уход на обед.

Желая отредактировать фрейм, если вы, например, продолжили работу над проектом после остановки *Watson*, можно запустить команду `watson edit id_фрейма`, которая открывает соответствующий фрейм в заданном текстовом редакторе, и затем можно изменить время окончания/начала. Можно также переименовать проект или тег командой `watson rename`.

Можете запустить команду `watson report -m`, чтобы посмотреть все проекты, над которыми работали в текущем месяце. Аналогичным образом можно создавать отчеты для определенного тега или проекта с использованием в отчетах параметров `-T` или `-p`:

```
$ watson report -T Roundup -m
```

```
Thu 01 February 2018 -> Thu 22 February 2018
```

```
LXF235 - 17h 01m 37s
```

```
[Сравнение 17h 01m 37s]
```

```
LXF236 - 9h 51m 36s
```

```
[Сравнение 9h 51m 36s]
```

```
Total: 26h 53m 13s
```

Здесь мы вставили параметры для создания отчетов по данному тегу за текущий месяц. Поскольку мы использовали один и тот же тег для разных проектов, *Watson* автоматически выдаст данные для обоих проектов.

Watson не распознает фреймы по имени проекта, значит, придется запустить команду `watson log` для списка тех команд *Watson*, которые вы уже запустили. Это предоставит вам список дат всех фреймов (последний идет первым) и соответствующее имя проекта/тега и т. д.

Ценность такого проекта, как *Watson*, осознать легко, даже если вам незачем хранить таблицы учета рабочего времени. Вас даже может удивить то, что он сообщит вам в конце рабочей недели. **LXF**

```
1: linuxlala@playground-budgie: ~
linuxlala@playground-budgie:~$ watson report -T Roundup -m
Thu 01 February 2018 -> Thu 22 February 2018
LXF235 - 17h 01m 37s
[Roundup 17h 01m 37s]
LXF236 - 9h 51m 36s
[Roundup 9h 51m 36s]
Total: 26h 53m 13s
linuxlala@playground-budgie:~$
```

Можно использовать автодополнение по Tab для имен команд, таких как `start` и `report`, а также для имен проектов и тегов.

СКОРАЯ ПОМОЩЬ

Если имя или тег проекта состоит из нескольких слов, следует заключать такие имена или теги в кавычки. Например: `+“вечерний чай”`.

Компиляция вдали: Разделим нагрузку

Матс-Таге Аксельссон охотно делится добром и экономит время на решение задач с высокой нагрузкой, скидывая часть нагрузки на другие компьютеры...



НАШ
ЭКСПЕРТ

Матс-Таге
Аксельссон

Когда Матс начал с Linux, у него был IBM Thinkpad с дискеттами. И он до сих пор пытается выжать максимум из стареющего оборудования, сквалыга!

СКОРАЯ ПОМОЩЬ

При тестировании установки выберите небольшой пакет на www.gnu.org/software, чтобы убедиться, что программа компилируется перед попыткой распределения нагрузки компиляции.

Ваша повседневная компьютерная деятельность обычно идет без сучка и задоринки, пока вы не заполните всю память и не будете вынуждены обратиться к подкачке. А если вы при этом компилируете программу или визуализируете графику, то мигом заметите неладное.

Оба эти вида деятельности интенсивно потребляют память и процессор, и это может занять много времени. Решение заключается в распределенных вычислениях: это метод использования нескольких компьютеров для выполнения одной задачи.

На нашем уроке мы сосредоточимся на том, как компилировать программные проекты на своих компьютерах. Мы также увидим, какие решения используют аниматоры для ускорения рабочего процесса. И изучим, в каких обстоятельствах необходима специальная настройка для кросс-компиляции.

Ваш первый шаг — установить *GCC*. Необходимые пакеты определяются в большинстве дистрибутивов пакетом *build-essential* или аналогичным.

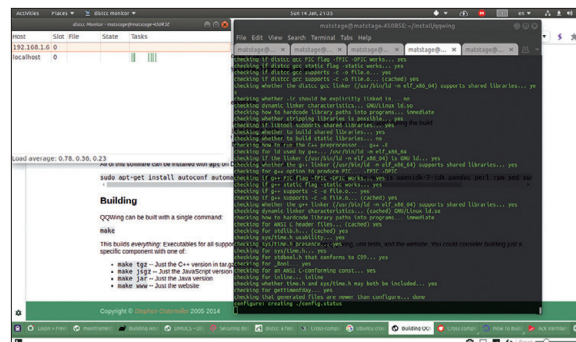
Поскольку *GCC* поддерживает многие языки помимо C или C++, доступно много дополнительных пакетов для программирования на других языках. Java — интересный пример, потому что его использует Android, но самым распространенным способом управления сборкой Java является Gradle. Здесь мы рассмотрим *gcc*, используемый для C и C++.

Сначала продемонстрируем, как выполнить компиляцию *GCC*: это достигается запуском команды *make*. Она войдет в привычку при компиляции своего или чужого кода. Главное теперь — обеспечить параллельную компиляцию. Команда *make* принимает параметр *jobs*, устанавливающий, сколько заданий может выполняться одновременно:

```
$ cd [your build directory]
$ make -j 4
```

Этому параметру надо задать значение в полтора–два раза больше числа ядер в вашей системе. Прежде чем вы сможете успешно распределить свою нагрузку, сначала ее надо распараллелить. Это позволит другим системам распределить ее на множество компьютеров.

Остальные параметры будут значительно отличаться для разных сборок и поэтому здесь не будут рассмотрены. При подсчете ядер необходимо учитывать, сколько ядер имеет каждый процессор в вашем «кластере». Параметр *jobs* иногда можно дополнить параметром нагрузки [*load*] — *-l*, определяющим максимальную нагрузку, которую должен ожидать компилятор перед запуском нового задания. Это полезно, если во время компиляции



Здесь *distccmon* показывает уровень активности в другом ЦП, и в этом случае использует *localhost* для компиляции тестового пакета.

вы хотите использовать компьютер для других задач, поскольку иначе он может затормозиться.

Общей проблемой надомных программистов является то, что у них есть только один компьютер. Лучшим способом избежать такого замедления работы является распределение ее на множество компьютеров. Один из способов сделать это — использовать пакет *distcc*, который примет команду *make* и распределит нагрузку согласно порядку, определенному вами в конфигурации. Пакет *distcc* доступен по умолчанию во многих дистрибутивах, в достаточно свежей версии. Ubuntu имеет версию 3.1, а последний RC — 3.2, выпущенный в 2014 г. Очевидно, что он староват, но пакет выглядит стабильным и удобным, несмотря на кажущуюся заморозку в разработке.

Установка *distcc* и перенаправление

distcc и его демон включены в стандартный пакет. При желании установить демон лучше всего скомпилировать его для конкретной платформы. Если у вас только права обычного пользователя, вы всё равно сможете установить и запустить *distcc*, однако зависимости, вероятно, будут проблемой, поэтому постарайтесь не делать этого!

```
$ sudo apt install distcc
```

Теперь, заполнив *distcc* на обоих или всех узлах, вы можете запустить демон на сервере, который также называется контрибьютор.

Демон слушает порт 3632, поэтому обеспечьте открытие этого порта для ваших контрибьюторов. Эта опция работает только для

TCP-соединения, и хотя оно быстрое, но не зашифровано, и поэтому должно использоваться только внутри вашей собственной сети. Как вам, несомненно, известно, всё передаваемое через Интернет должно быть защищено.

При использовании SSH клиент *distcc* сам вызовет демона. Перед началом компиляции вам следует решить, какие хосты использовать для заданий, которые вы хотите выполнить. Это делается очень просто: в *.distcc/hosts* список будет содержать хосты, куда вы хотите отправить компиляцию. В режиме SSH перед именами хостов будет знак @. У вас также есть возможность использовать *Avahi*, также известный как *zeroconf*, чтобы отыскать хосты:

```
localhost 192.168.1.6
```

Он же при использовании SSH:

```
@localhost @192.168.1.6
```

После настройки надо перейти в каталог с исходниками и настроить компиляцию в соответствии с инструкциями пакета. Перед запуском *make* нужно добавить *distcc* или *rump* в зависимости от необходимой вам функциональности. Для использования любой из удаленных функций нашего урока добавьте нужную вам систему в переменную окружения *CC*:

```
$ export CC="distcc gcc"
```

С этой переменной команда *make* будет использовать команду *distcc* и распределять наилучшим образом. Это хорошо работает и для других систем сборки, если они совместимы с цепочкой инструментов *GCC*. Установка переменной окружения *CC* — это только один способ заставить *distcc* компилировать проекты. У этого метода есть проблема: дело в том, что многие компиляторы предполагают, что в переменной *CC* нет пробелов. По этой причине наиболее распространенный метод — метод маскировки (и тот, что используется вашим менеджером пакетов) — это установка ссылок в каталоге */usr/lib/bin*. Чтобы убедиться, что маскировка работает, надо располагать каталог в начале пути, поэтому обычный *GCC* не запускается нераспределенным.

Перед началом компиляции надо сконфигурировать проект, и обычно для этого используется *autoconf*. Задача *autoconf* не распределяется *distcc*, потому что случай для предварительной обработки с *make* отличается.

Пока что *distcc* не будет распределять предварительную обработку для вас по умолчанию. Для этого вам необходимо активировать режим "rump". Но это сделать легко — просто добавьте *rump* перед командой *make*:

```
$ sudo rump make -j8
```

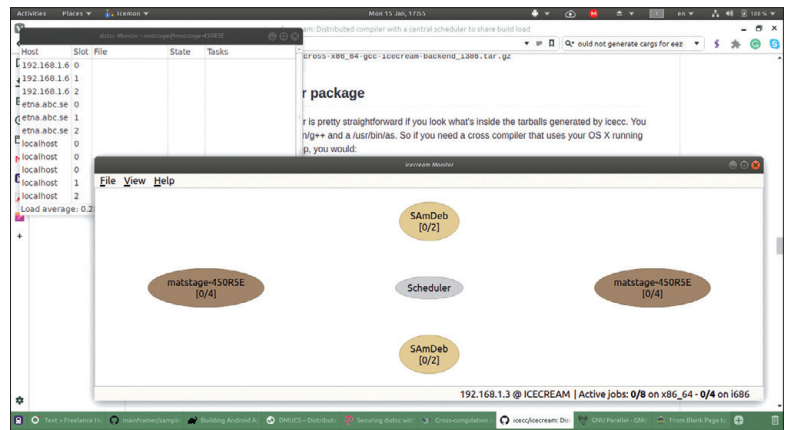
Учтите, что мы выполняем эту компиляцию от имени *root*. Для большинства пакетов это не является строго необходимым, однако некоторые компиляции иначе не сработают. В остальном *root* необходим только при установке на машину, на которой вы находитесь.

В данном случае у автора есть много разных платформ, идеально подходящих для компиляции. Старый нетбук, хотя и маломощный, еще найдет применение. Тогда возникает проблема, что стандартный компилятор будет установлен на платформу, на которой он компилируется. В этом случае нетбук 32-, а новый компьютер — 64-разрядный.

Значит ли это, что наши старые 32-битные машины бесполезны? Конечно, нет, потому что для таких случаев есть кросс-компиляция или поддержка *multi-lib*. Эти библиотеки позволяют собирать на любой платформе (ну, почти) и с любой платформы. Наиболее распространенный подход заключается в компиляции на мощных серверах для встроенных систем и небольших машин, таких как Raspberry Pi.

Чтобы установить правильные пакеты, найдите *gcc-multilib* или *g++-multilib* вашим менеджером пакетов и установите необходимые библиотеки следующим образом:

```
$ sudo apt install gcc-multilib
```



Здесь монитор *i9cc* имеет улучшенный интерфейс и может показать вам много разных аспектов процесса компиляции.

После установки пакета *multi-lib* обычно добавляется еще одна библиотека, в данном случае — *amd64*. Убедитесь, что платформа, для которой вы хотите компилировать, установлена так, чтобы они и файлы заголовков были доступны при повторной попытке.

Если вам нужны библиотеки для большего количества платформ, выполните поиск с помощью *apt*. Примером может служить следующее:

```
$ apt search libgcc-19pre1 arm
```

Apt предупреждает вас, что он не очень хорошо обрабатывает CLI, но это имеет значение только при написании служебных скриптов, так что можно спокойно игнорировать подобные предупреждения.

Обеспечив все узлы всем необходимым программным обеспечением, вы должны установить, какой из них использовать первым. Это важно, поскольку в первую очередь лучше выбирать самую быструю машину. Для этого в список файла хостов *distcc* поместите узел первым. Если у вас много одинаковых серверов, можно добавить флаг *--randomize*, чтобы делать выбор случайным образом.

Если у вас много машин, то хорошая идея — исключить *localhost*. Также обратите внимание, что при бое *distcc*

СКОРАЯ ПОМОЩЬ

Если вы пишете программу на основе GNU, подайте заявку на учетную запись на ферме компиляции GNU. Вас могут допустить, если вы вносите вклад в дело: <https://gcc.gnu.org/wiki/CompileFarm>.

» СОЗДАЙТЕ СВОЮ РЕНДЕР-ФЕРМУ

При расширении проекта набирается много пользователей, помогающих с тем же или даже с другим кодом. В таких случаях может потребоваться совместное использование вычислительной мощности всех машин: и ваших, и ваших друзей и коллег. В конце концов, полностью вы задействуете свой компьютер только при компиляции, рендеринге графики и в насыщенных графикой играх.

Поскольку вы уже используете *distcc*, самым простым решением является установка *DMUCS* (распределенная многопользовательская компиляция). При настройке системы и подключении всех хостов обеспечьте, чтобы они могли запускать *distcc* и устанавливать приоритета в соответствии с вычислительной мощностью узла. Система разработана так, что многие пользователи могут компилировать одновременно; она хорошо масштабируется для обработки различных нагрузок. Она также использует все ядра и балансирует нагрузку, чтобы все процессоры использовались без перегрузки машины, где они установлены.

Машина, где пользователь запускает процесс, требующий высокой нагрузки, сообщает об этом планировщику, и отправки нагрузки на нее будут уменьшены. Это означает, что вы можете использовать все компьютеры команды — даже если они в это время занимаются обычными задачами — не прерывая чью-либо работу. Хосты также можно динамически удалять из фермы.

компиляция продолжается на localhost по умолчанию, даже если вы не определили его в вашем файле `.distcc/hosts`.

Желая увидеть, что происходит с различными вашими помощниками, вы можете установить и запустить `distccmon`. Он поставляется и для Gnome, и как текстовый монитор. Его параметры — `-text` или `-gnome`, по умолчанию — в Gnome.

```
$ distccmon
```

Интерфейс представляет собой простой список с графическим представлением загрузки хостов. В одной строке вы видите хост, текущую активность и, наконец, гистограмму текущих заданий на сервере.

Извлекь из Iccss больше

`Iccss` имеет ряд функций, в определенных случаях полезных. Одна из них улучшает планирование и может управлять помощниками (но тогда требуется центральный сервер). Есть также функция попроще, для переключения версии компилятора и лучшей поддержки кросс-компиляции.

Обычно в домашней обстановке, где у вас есть тот же компьютер и тот же дистрибутив, вы можете также придерживаться `distcc`, но если они перемешаны, то стоит рассмотреть `icccream` и `icccs`. Когда у вас много машин, вам потребуется `icccsd -d` для большинства из них и `icccs-scheduler -d` — по крайней мере, для одной из них. Обратите внимание, что ради резервирования можно настроить планировщик на нескольких компьютерах: когда один планировщик выходит из строя, другой автоматически включается.

Планировщик будет отправлять задания, учитывая скорость узлов и обеспечивая максимальную эффективность компиляции. Если вам нужен дополнительный стимул для использования `icccs` вместо `distcc`, рассмотрите графический инструмент мониторинга `icemon`. Он предоставляет шесть разных видов и гораздо более привлекательную графику, иллюстрирующую состояние задач компиляции.

Чтобы насладиться всеми преимуществами вашего компьютера (и при наличии места на диске), добавьте в настройку `ccache`. Это лучше всего работает, когда приходится регулярно перекомпилировать, поскольку `ccache` потом выбирает части, которые не были изменены. Распространенный пример — добавление небольшого исправления в крупный проект. Если вы используете `ccache`, то система будет компилировать только затронутые части. Это может быть только код, который вы изменили; конечно, часть используемого кода тоже может быть задействована, но в итоге эффект обычно очень заметный.

Сколько дискового пространства вам понадобится, будет зависеть от компилируемого проекта, но, в качестве примера для сравнения, для компиляции `Firefox` требуется около 20 ГБ

на диске. Зависит от вас, хотите ли вы смонтировать отдельный раздел. Лучший подход — использовать отдельный SSD, благодаря скорости, которую вы можете получить от него. Затем вы решите, куда его поместить в файле конфигурации. Автор подключился к `/mnt/ccache`, а затем использовал `matstage` в качестве пользователя, создав тем самым свой собственный каталог.

```
$ sudo mkdir /mnt/ccache/matstage
```

```
$ sudo chown matstage:matstage /mnt/ccache/matstage
```

Настройки в файле `~/.ccache/ccache.conf` —

```
max_size = 20G
```

```
cache_dir = /mnt/ccache/matstage
```

```
prefix_command=icccc
```

Для запуска `icccs` с `ccache` мы добавили оператор `prefix_command` в конфигурацию `ccache`. Для тестирования добавьте следующие параметры в свой скрипт конфигурации; `--with-ccache` и `-j8` в свой скрипт `make`. Итак, для `Firefox` это будет `ac_add_options --with-ccache=/usr/bin/ccache`

```
mk_add_options MOZ_MAKE_FLAGS="-j12"
```

Каждый проект немного отличается, но оба варианта, как правило, доступны в той или иной форме, и вы всегда можете добавить его в командной строке.

Освойте Gradle

Если вы следили за нашим предыдущими учебниками по Android, то знакомы с использованием `Gradle` для компиляции ваших проектов. Может показаться, что `Gradle` подходит только для компиляции проектов Android. Однако на самом деле `Gradle` является системой сборки, которую многие разработчики используют для компиляции широкого диапазона проектов и языков программирования.

Когда вы хотите скомпилировать свой проект с `Gradle`, подумайте об использовании `mainframer`. Это расширение позволяет запускать компиляцию на другом сервере — после правильной настройки. Процедура настройки начинается с обеспечения доступа к удаленному узлу (узлам) через SSH и хорошей связи с ними.

Затем надо скопировать скрипт в корень вашего проекта. `Mainframer` — это скрипт оболочки, что делает его достаточно простым для запуска и изменения. Для скрипта требуется файл настройки в каталоге `.mainframer` в корне вашего проекта.

Содержимое файла настройки уже включает параметры удаленной машины и уровня сжатия, и требуется только объявление удаленного компьютера:

```
remote_machine=Samdeb
```

```
local_compression_level=4
```

```
remote_compression_level=4
```

Другие файлы в каталоге `.mainframer` включают правила игнорирования, которые работают так же, как и правила `rsync`. Причина в том, что `rsync` — это пакет, используемый для передачи файлов. В данном примере проект небольшой, и вы можете безопасно пропускать файлы игнорирования и двигаться дальше. При первом запуске скрипт будет искать Android Studio, в том числе проверку того, приняты ли вы лицензию, и того, что у вас есть правильный NDK.

Далее, на удаленных машинах у вас может не быть оконного менеджера, из-за чего установка всей Studio становится ненужной, а то и невозможной. Решение состоит в том, чтобы обходиться только инструментами командной строки. Их можно загрузить в виде ZIP-файла с сайта Android:

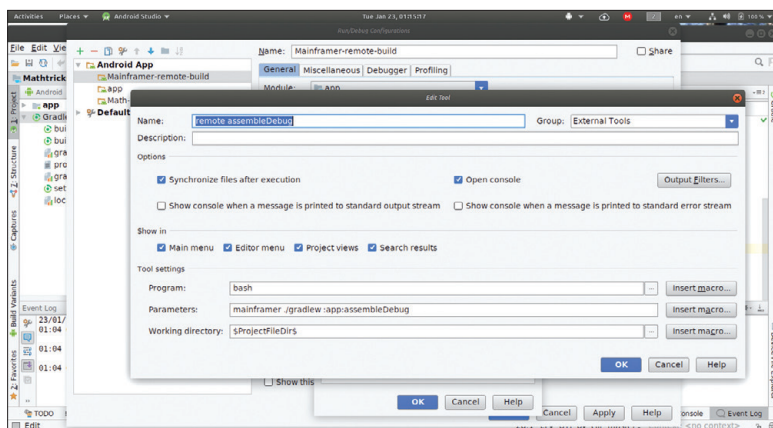
```
$ wget http://dl.google.com/android/repository/sdk-tools-linux-3859397.zip
```

Загрузив инструменты, вы должны распаковать их в каталог `Android/Sdk`, где по умолчанию работает всё программное обеспечение для разработки Android.

Подтверждение того, что вы приняли лицензию, находится в каталоге лицензий в библиотеке SDK. Если на удаленной

СКОРАЯ ПОМОЩЬ

Для параллельного выполнения любой коллекции команд может использоваться GNU Parallel, если, конечно, команды не зависят от результата друга. Подробнее — на www.gnu.org/software/parallel/.



В Android Studio не забудьте правильно установить параметры в меню конфигурации запуска/отладки, потому что в путях к двоичным файлам легко сделать ошибку.

машине вы устанавливаете только средства командной строки, то единственным способом принять лицензию является копирование лицензионного каталога с одной машины на другую. Простейший способ сделать это — применить *sftp*:

```
$ cd Android/
$ sftp samdeb
$ cd Android/
$ put licenses/*
```

Без файлов лицензий *Gradle* не сможет загружать пакеты платформы в соответствии с вашим проектом. При первом запуске соответствующие файлы платформы будут загружены на удаленный компьютер. Система также будет искать NDK — если в вашем проекте нет кода на C, можете игнорировать это предупреждение..

Без файлов лицензий *Gradle* не сможет загружать пакеты платформы в соответствии с вашим проектом.

Желая запустить инструмент с использованием локальной версии, используйте команду

```
$ ./mainframer ./gradlew build
```

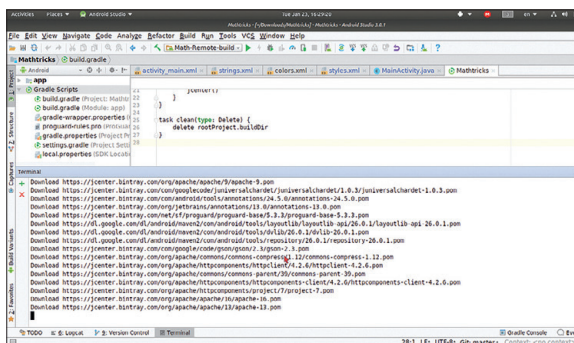
Основная часть пакета *mainframer* должна быть добавлена в файлы конфигурации проектов в корне. Чтобы активировать его, в меню Run > Configure найдите раздел Before build. В этом разделе надо удалить запись 'Gradle aware': подключение к *Gradle* будет обрабатывать *mainframer*. В квадрате вы устанавливаете 'bash' и 'mainframer gradlew...', чтобы *mainframer* управлял работой *Gradle*. После этого *mainframer* будет запускать задания на указанном вами узле для обработки. В настоящее время в этом пакете нет кластерной функциональности, это просто способ переноса заданий на более мощный сервер.

Другие задачи с высокой нагрузкой

К счастью, компиляцию программного обеспечения легко разделить на множество задач и распределить нагрузку. Говоря «легко», мы имеем в виду, что процесс состоит из многих независимых задач, которые поддаются распределению.

Другая хорошо распределяемая задача — это рендеринг графики. Механизм рендеринга *Cycles* в *Blender* имеет функцию планирования, и мы смотрим, как сделать и в то же время сделать это эффективно. Лучший способ сократить время выполнения задания — это распределить задачи с помощью планировщика.

Планировщик, созданный для *Cycles*, назван *Flamenco*, и, как вы уже догадались, он исходит от команды *Blender* в Амстердаме. Он включен сюда для сравнения, а также потому, что команда утверждает: вы можете использовать этот пакет для распределения другой работы на множество компьютеров. Это требует некоторых усилий в разработке, но всё же это интересная перспектива.



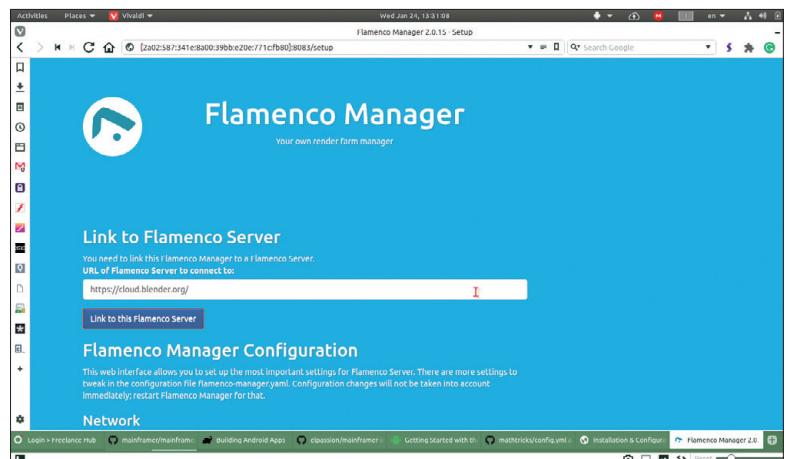
Менеджер визуализации *Flamenco* создан для работы с облаком *Blender*, но его можно настроить для работы на ваших личных ПК. Хотя пробуйте чисто на свой страх и риск!

»» ОНЛАЙН-ФАБРИКИ КОМПИЛЯЦИИ

В Интернете множество сообществ и компаний, бесплатно обеспечивающих мощности для онлайн-компиляции для проектов с открытым исходным кодом. Фактически, многие из них тоже имеют бизнесы, которые платят большие деньги за компиляцию собственных проектов; но частные лица, использующие свободные лицензии, имеют несколько свободных проектов на выбор. Обычно вам надо сохранить свой проект на сайте программного обеспечения, таком как Github или аналогичный.

Сервис называется Constant Integration, а известные сайты включают Travis CI и circleCI. В обоих случаях вы можете использовать Github для хранения, добавить кое-какую конфигурацию в свой проект, и оба сайта скомпилируют его для вас. Отчеты об ошибках остаются на сайте для изучения и исправления до тех пор, пока не будет достигнута успешная компиляция.

Чтобы начать работу, войдите в свою учетную запись Github и добавьте свой проект на сайт. На circleCI вы добавляете короткий файл, задающий язык программирования, на котором написан проект. Для использования этих сервисов вам необходимо изучить основы YAML. Чтобы создать собственный сайт для сборки, есть также Buildbot, который можно установить на вашей обычной машине Linux с помощью *Docker*. Учебник на www.buildbot.com прост для изучения, хотя получить от него пользу в полной мере — это дело другое...



При первом запуске *mainframer* загрузит все необходимые библиотеки на удаленный узел и оставит их в каталоге кэша. Вам надо убедиться, что места на дисках хватает.

Онлайн-движки для ML

Если у вас есть модель, которую вам надо обучать для вашего проекта машинного обучения (ML), вам, возможно, придется долго ждать накопления достаточного количества проходов по собранным вами данным. Наилучшим способом при больших наборах будет использовать облачные решения. Некоторые из них включают GCP, Amazon и Alibaba. Google Cloud Platform готова хранить ваши файлы, искать ваши данные — и да, обучать ваши модели. Надеемся, что в скором времени мы сможем использовать квантовые компьютеры, чтобы помочь с этим; они будут абсолютно превосходны для выполнения этой работы.

На нашем уроке мы увидели, что вы можете употребить свою старую машину для выполнения задач, которые раньше на одной машине отнимали много времени, и ничто не может этого изменить. Мы исследовали много экономящих время альтернатив, поэтому в следующий раз, когда вы вынуждены ждать, взвесьте свои альтернативы и уменьшите шансы на отвлечение внимания! **LXF**

»» ПОДПИШИТЕСЬ НА ЖУРНАЛ **LINUX FORMAT** на www.linuxformat.ru/subscribe/!

NEXTCLOUD

Интегрируемся с ONLYOFFICE

Владимир Диаконов устанавливает и настраивает облачный сервис с интегрированным офисным онлайн-пакетом на собственном сервере.



**НАШ
ЭКСПЕРТ**

Владимир Диаконов — адвокат, его специализация — юридическое сопровождение бизнеса. Он сумел научить пользоваться Linux даже своих пожилых родственников.

Облачные технологии стали ныне обычным атрибутом работы корпоративных систем. Они используются по различным направлениям. Сейчас предлагается много облачных сервисов — как бесплатных, так и платных — с различной функциональностью. Однако при использовании чужими облачными сервисами не гарантирован полный контроль над своими данными. Если на чужом облачном сервисе случится авария, то повлиять на эту ситуацию будет невозможно.

Решение данной проблемы заключается в том, что на сегодняшний день многие имеют свои собственные серверы или арендуют их. В повседневной деятельности такие серверы используются для обеспечения пользователям совместного доступа к информации, используемой в бизнес-процессах, и для работы с этой информацией.

Следовательно, если уж имеется сервер, в который вложены определенные денежные средства, то его надо использовать по максимуму и выжать из него всё, что возможно. В таких случаях сервер еще целесообразно использовать для хостинга своих корпоративных сайтов, для размещения сервера электронной почты и, наконец, для размещения облачных сервисов.

Современные Linux-серверы позволяют активно применять технологии виртуализации, что позволяет значительно повысить в бизнесе эффективность использования корпоративных серверов на ОС Linux.

В данной статье предлагается корпоративное решение на базе сервера Ubuntu 18.04 LTS, включающее в хостинг сайтов с панелью управления хостингом ISPConfig 3.1, набор облачных сервисов Nextcloud, включающих корпоративное облачное хранилище данных, корпоративный календарь, корпоративную электронную почту, интегрированный в Nextcloud сервер обработки документов ONLYOFFICE Document Server, позволяющий создавать и редактировать, в т. ч. совместно, текстовые документы и электронные таблицы.

Функциональность Nextcloud и ONLYOFFICE можно значительно расширить за счет подключения широкого спектра плагинов. В частности, в данном случае возможно создание полноценной CRM-системы, если таковая требуется. CRM — Customer Relationship Management — переводится как «Система управления взаимоотношениями с клиентами». Однако ни одна из предлагаемых на рынке CRM-систем онлайн не предлагает облачного хранилища документов и онлайн-сервиса для совместной работы над документами. И в нашем конкретном случае всё это физически расположено на собственном сервере, что позволяет обеспечить резервное копирование информации и полный контроль над ней.

ONLYOFFICE, как указано на сайте проекта <https://ONLYOFFICE.com>, это проект команды специалистов из Ascensio System SIA, IT-компании с головным офисом в Риге (Латвия). Однако, согласно данным на сайте, имеется представительство компании в Нижнем Новгороде.

Сегодня офисный пакет ONLYOFFICE набирает популярность. Одной из причин популярности является — тут скрывать нечего — очень высокая совместимость с форматами *Microsoft Office*. Популярности ему прибавляет то, что данный продукт в базовых версиях распространяется по лицензии GPL. Недаром во многих комментариях к релизу Ubuntu 18.04 авторы этих комментариев рекомендуют установить локальный офисный пакет ONLYOFFICE сразу после инсталляции Ubuntu 18.04.

В свою очередь, решение по интеграции позволяет установить Сервер документов на локальном сервере и интегрировать онлайн-редакторы с Nextcloud.

Сам сервер документов — это офисный пакет для работы онлайн, который включает просмотрщики и редакторы текстовых документов, электронных таблиц и презентаций, полностью совместимые с форматами Office Open XML (DOCX, XLSX, PPTX) и позволяющие совместно редактировать документы в режиме реального времени.

Сервер документов ONLYOFFICE включает:

- Редактор документов
- Редактор электронных таблиц
- Редактор презентаций
- Приложение «Документы» для iOS
- Совместное редактирование
- Поддержка иероглифов
- Поддержка всех популярных форматов: DOC, DOCX, TXT, ODT, RTF, ODP, EPUB, ODS, XLS, XLSX, CSV, PPTX, HTM.

Имея на сервере Nextcloud с интегрированным сервером документов ONLYOFFICE, можно при помощи любого устройства, где есть веб-браузер, работать с документами, хранящимися в Nextcloud. Данная связка является, на мой взгляд, в наибольшей степени гибким решением для бизнеса: она позволяет вести совместную работу над документами и бизнес-проектами и обеспечивает надежную сохранность данных путем синхронизации данных на сервере с данными на локальных рабочих станциях.

А теперь мы с вами перейдем непосредственно к развертыванию нашей системы.

Я исхожу из того, что на сервере уже установлена и сконфигурирована ОС Ubuntu Server 18.04 LTS с LAMP (как вариант, можно любой другой), почтовым сервером Postfix и панелью управления хостингом ISPConfig 3.1. Каким образом всё это устанавливается

и конфигурируется, писать здесь нет никакого смысла, т. к. на эту тему великое множество материалов в Сети.

Итак, наш сервер запущен. Мы имеем имя домена второго уровня, например, `test.ru` (это условное доменное имя, далее используемое в статье), на котором разместится наш корпоративный сайт. Для размещения облачного сервиса Nextcloud мы будем использовать доменное имя третьего уровня `nextcloud.test.ru`. Можно взять и любое другое доменное имя третьего уровня, покороче, чем `nextcloud`: просто так будет лучше для удобства восприятия. Управление данным сайтом на нашем сервере мы осуществляем через ISPConfig 3.1. Как это делается, подробно описывать в данной статье также нет смысла, ввиду изобилия материалов в Сети на эту тему.

Отмечу лишь, что в терминологии панели управления хостингом ISPConfig 3.1 наш сайт `test.ru` относится к клиенту № 1 и является у этого клиента вторым сайтом. Следовательно, на Ubuntu Server 18.04 LTS файлы ресурса `test.ru` физически расположены в его корневом каталоге — `/var/www/clients/client1/web2/web/`.

Для работы с дистрибутивом Nextcloud перейдем к режиму работы от имени `root`:

```
sudo su
```

Вводим пароль `root` и переходим в корневой каталог `test.ru`:

```
cd /var/www/clients/client1/web2/web/
```

Загружаем с сайта `nextcloud.com` свежую версию дистрибутива Nextcloud:

```
wget https://download.nextcloud.com/server/releases/nextcloud-13.0.4.zip
```

Распаковываем архив с дистрибутивом Nextcloud в корневой каталог ресурса `test.ru`, `/var/www/clients/client1/web2/web/`:

```
unzip nextcloud-13.0.4.zip -d/var/www/clients/client1/web2/web/
```

По завершении этой операции у нас в корневом каталоге ресурса `test.ru` должен появиться каталог с дистрибутивом Nextcloud. Именно в этом каталоге будет в дальнейшем размещаться сам облачный сервис Nextcloud.

Для успешной установки Nextcloud необходимо изменить владельца каталога `nextcloud`:

```
chown -R www-data:www-data /var/www/clients/client2/web4/web/nextcloud/
```

и установить права доступа к каталогу `nextcloud`:

```
chmod -R 755 /var/www/clients/client1/web2/web/nextcloud/
```

Теперь надо создать для Nextcloud базу данных `MySQL`. Это можно сделать любым удобным способом: через `phpmyadmin`; через панель управления хостингом ISPConfig 3.1; через командную строку. Выберем последний способ. Мне он импонирует значительно большей быстротой и надежностью работы с особенно большими дампами баз данных. Создание базы данных — не очень ресурсоемкая операция, но всё же, на мой взгляд, хороший стиль администрирования — это командная строка.

Заходим в `MySQL` через командную строку:

```
mysql -u root -p
```

Система запросит пароль `root`; введем его.

Далее будем создавать базу данных для Nextcloud. Для корректной работы база данных должна иметь кириллическую кодировку `UTF-8`. Для реализации этих условий выполняем команду создания базы данных `MySQL`:

```
mysql> CREATE DATABASE c2nx CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Здесь `c2nx` — имя создаваемой базы данных.

Создаем для этой базы данных пользователя, который будет с ней работать:

```
mysql> GRANT ALL PRIVILEGES ON c2nx.* TO c2nx@localhost IDENTIFIED BY 'пароль_пользователя';
```

Теперь, в принципе, можно запустить установку Nextcloud, набрав в адресной строке адрес `test.ru/nextcloud/`.

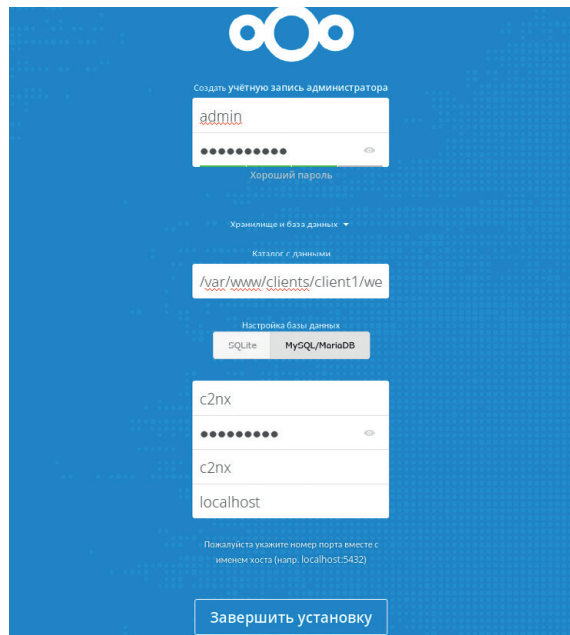


Рис. 1. Установка параметров инсталляции Nextcloud.

Появится главная страница Nextcloud. Нам предложат создать учетную запись администратора. В поле логин вводим «admin», в поле пароля вводим пароль администратора.

Потом необходимо ввести учетные параметры базы данных. Для этого жмем стрелочку рядом с надписью «Хранилище и база данных». В раскрывающихся полях уже видим путь к каталогу, в котором лежит Nextcloud: `/var/www/clients/client1/web2/web/nextcloud/`. Здесь ничего не трогаем.

Под надписью «Настройка базы данных» выбираем пункт `MySQL/MariaDB` и заполняем поля с данными для доступа к базе данных:

- Пользователь базы данных: `c2nx`.
- Вводим пароль для доступа к базе данных для пользователя `c2nx`.
- Вводим название базы данных `c2nx`.
- Оставляем хост базы данных: `localhost`.
- Нажимаем `Завершить установку`.

Выглядеть это должно как на рис. 1.

После этого установка Nextcloud будет завершена и загрузится корневой каталог облачной системы.

Но для солидности надо создать доменное имя третьего уровня: `nextcloud.test.ru`, чтобы Nextcloud запускался при вводе в адресную строку именно этого имени. Для этого зайдём в панель управления хостингом ISPConfig 3.1. У меня это адрес `https://server1.test.ru:8080/`. После авторизации выбираем в меню Сайты, затем в левом меню пункт Поддомен, и попадаем в пункт создания поддомена.

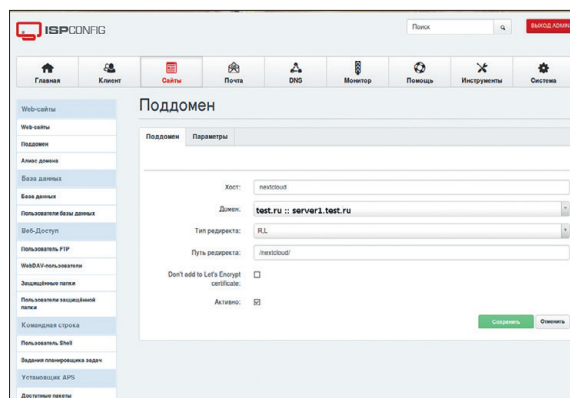
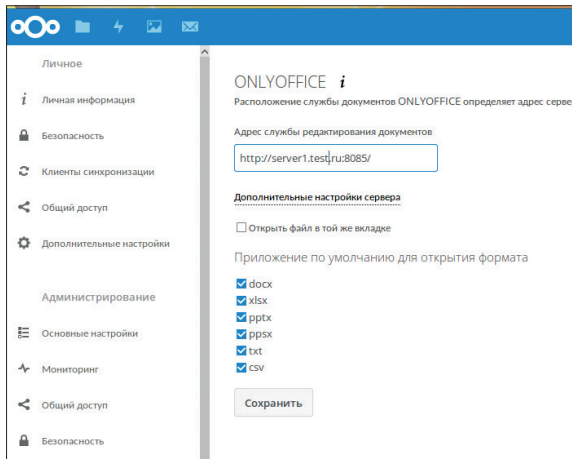


Рис. 2. Установка параметров поддомена 3-го уровня `nextcloud.test.ru`.

Рис. 3. Параметры интеграции Nextcloud и ONLYOFFICE.



В графе Хост вводим "nextcloud", в пункте Домен из выпадающего списка выбираем домен "test.ru: server1.test.ru", в пункте Тип редиректа из выпадающего списка выбираем тип редиректа "R, L", в пункте Путь редиректа вводим путь редиректа при наборе адреса nextcloud.test.ru: "/nextcloud/". Ставим галочку в пункте Активно. Нажимаем Сохранить. Настройки поддомена должны выглядеть как на рис. 2.

Теперь приступим к установке онлайн-редактора ONLYOFFICE и его интеграции с Nextcloud.

Для установки сервера документа воспользуемся *Docker*. Дело в том, что это разновидность виртуализации, позволяющая изолировать зависимости приложений, тем самым избегая возможного конфликта между приложениями и вдобавок обеспечивая легкое и быстрое развертывание приложения благодаря контейнерам и существенно меньшее потребление ресурсов компьютера, чем обычные виртуальные машины. Видимо, по этой причине недавно презентован дистрибутив Ubuntu 18.04 LTS Minimal Server, один из вариантов которого предназначен именно для *Docker*. Эти аспекты обусловили в данном случае для установки ONLYOFFICE Document Server применение *Docker*.

Сначала установим *Docker* на Ubuntu Server 18.04 LTS:

```
apt install docker.io
```

Запустим *Docker* и сделаем его автоматически загружаемым при старте системы:

```
systemctl start docker
```

```
systemctl enable docker
```

Теперь можно устанавливать ONLYOFFICE Document Server. Перед этим следует определить, через какой порт будет осуществляться доступ к ONLYOFFICE Document Server. Как правило, порт 80, 8080 на серверах всегда задействованы. Значит, надо выбрать какой-нибудь другой порт. Выберем порт 8085.

С портом определились; вводим теперь команду на установку ONLYOFFICE Document Server в терминале (напомним, во всей этой цепочке операций мы работаем как root):

```
docker run -i -t -d -p 8085:80 --restart=always ONLYOFFICE/documentserver
```

По этой команде происходит установка ONLYOFFICE Document Server со всеми зависимостями. Объем загрузки порядка 2 Гб.

По окончании загрузки вводим в адресную строку браузера адрес <http://server1.test.ru:8085/>; загрузится web-страница с сообщением, что ONLYOFFICE Document Server работает [Document Server is running].

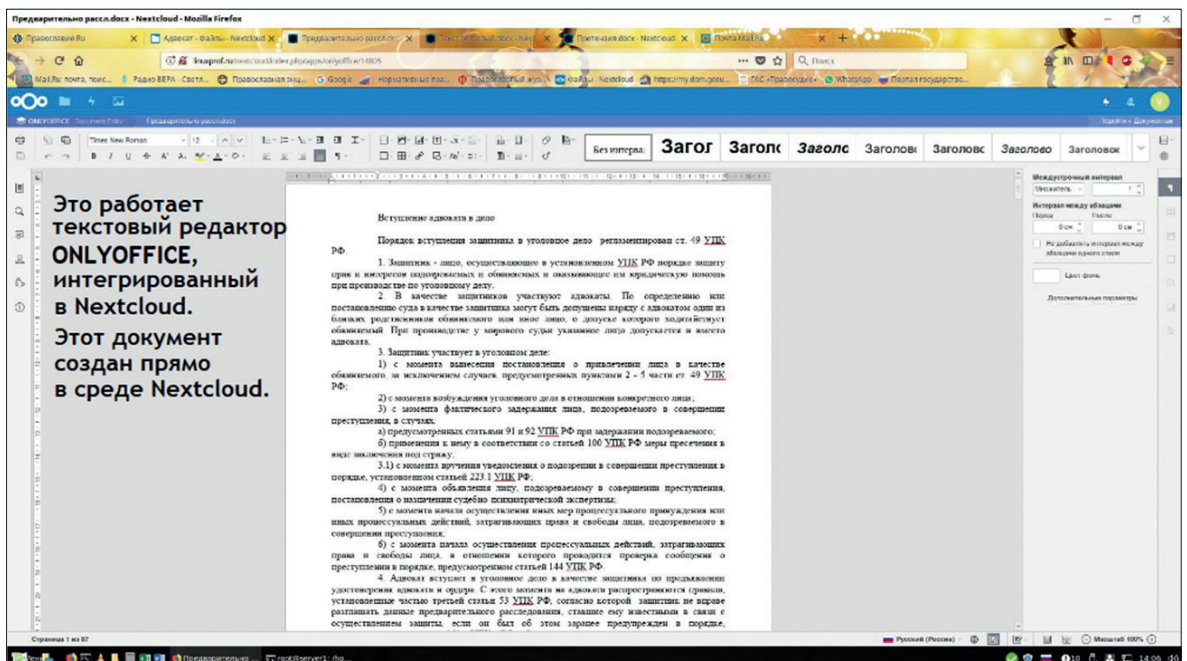
Теперь нам осталось интегрировать ONLYOFFICE Document Server и Nextcloud. Для этого заходим в установленный нами Nextcloud через браузер: <http://nextcloud.test.ru/> под учетной записью администратора.

Щелкаем мышью по иконке учетной записи в правом верхнем углу, и из выпадающего меню выбираем пункт Приложения; открывается страница приложений. В левой колонке выбираем категорию Офис и текст, находим в этой категории приложение ONLYOFFICE и нажимаем под ним Включить.

Далее надо перейти опять в меню учетной записи администратора и нажать первый пункт меню «Настройки». В центральной части страницы мы видим личные настройки учетной записи, а нам нужна левая часть. В столбике находим пункт ONLYOFFICE и нажимаем на него. В настройках ставим галочки для всех типов файлов. Самая главная настройка — это Адрес службы редактирования документов. В него мы прописываем адрес Document Server: <http://server1.test.ru:8085/>. Больше ничего не трогаем и нажимаем Сохранить (рис. 3).

Всё, мы развернули облачный сервис Nextcloud с интегрированным офисным онлайн-пакетом ONLYOFFICE (рис. 4). Теперь в Nextcloud можно создавать и редактировать текстовые документы, электронные таблицы и презентации, совместимые с форматами Open Document и *Microsoft Office*. LXF

Рис. 4. Работает текстовый редактор ONLYOFFICE, интегрированный в Nextcloud.





SUSE Linux Enterprise Server 11

SUSE Linux Enterprise Server 11 Service Pack 3 — высоконадежная, масштабируемая и безопасная серверная операционная система для предоставления IT-услуг корпоративного уровня в физической и виртуальной инфраструктурах. Операционная система создана для работы в смешанных окружениях и предлагает лучшую в своем классе производительность.



Сертифицировано ФСТЭК

Сертифицированная ФСТЭК операционная система SUSE Linux Enterprise Server 11 Service Pack 3 является программным средством защиты от несанкционированного доступа к информации, не содержащей сведения, составляющие государственную тайну, и соответствует требованиям руководящих документов «Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации» (Гостехкомиссия России, 1992) — по 5 классу защищенности и «Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недеklarированных возможностей» (Гостехкомиссия России, 1999) — по 4 уровню контроля.

Операционная система разработана в соответствии с ТУ 501110-007-82487552-14.

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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



ОБЛАЧНОЕ ХРАНЕНИЕ

Синхронизация и шифрование данных в облаке

Брайан Морк открывает, как заложить основу надежного и свободного облачного хранилища, которое шифруется с помощью *Syncthing+* и *Encfs*.



НАШ ЭКСПЕРТ

Брайан Морк использует Linux 23 года, с тех пор, как в 1994 г. установил Slackware, и написал на эту тему десятки статей.

Облачное хранение файлов становится всё популярнее с каждым днем. Однако коммерческим решениям не хватает конфиденциальности, они небезопасны, лишают контроля за данными, часто требуют установки инвазивного ПО и навязывают потребителю модель приобретения по подписке.

В этой серии из двух уроков мы обсудим архитектуру для создания вашей собственной бесплатной резервной копии в облачном хранилище, шифрующейся как во время передачи данных, так и когда они пребывают в покое. Обычно вы можете разместить резервную копию на дружественном удаленном сайте, например, в доме друга. ПО является совместно используемым и доступно для скачивания из Интернета. Спецификации оборудования включают либо старый компьютер, который где-то у вас пылится, либо Pi 3 за £25 вместе с постоянным USB-дискон, емкость которого отвечает вашим запросам.

Компьютерная среда

Спрос на сервисы облачного хранения не нов, и в Сети полно разрозненных частей этого решения:

- ▣ Коммерческие облачные сервисы, такие как Google Drive, Amazon S3, Microsoft OneDrive (или Azure), Dropbox, SpiderOak, CrashPlan, pCloud, Cloudup, Tresorit, CloudApp и Duplicr.
- ▣ Домашние решения, такие как Tonido, ownCloud, FileCloud, Pydio, Sparkleshare, Seafile, Box, 4Shared, Yandex.Disk и SugarSync.

▣ Коммерческие P2P-методы, такие как Storj или Sia, распределяют ваши зашифрованные файлы по совместной подборке узлов-участников, что позволит вам покупать и продавать место на диске.

▣ Синхронизировать файлы можно с помощью *BTSync* (теперь *Resilio*), *Syncthing* (и ответвлением *Pulse*), *Syncany*, *Librevault*, *Koofr*, *Unison*, *AeroFS*, *TeamDrive*, *CmisSync*, *Tahoe-LAFS*, *mintstash* и т. д.

▣ Файлы можно шифровать с помощью *Veracrypt* (переписанным *TrueCrypt*), *EncFS*, *CryFS*, *ecryptfs*, *GNU Privacy Guard*, *dm-crypt*, *CipherShed* (ответвлением *TrueCrypt*), *Go-CryptFS* или множества других.

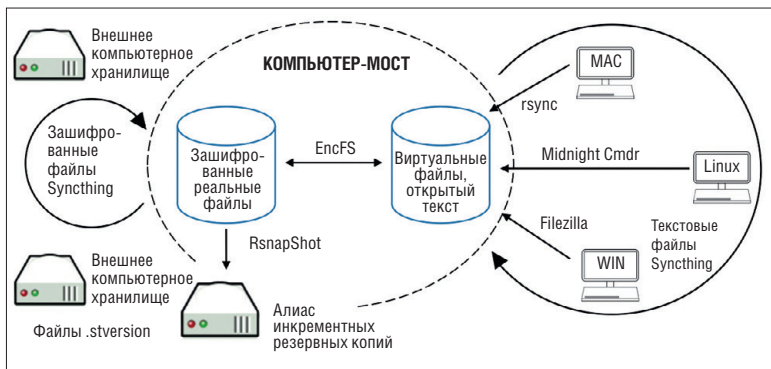
В какофонии вышеперечисленных опций вы сможете сделать правильный выбор, если точно сформулируете свои требования. Итак, вы можете решить, что вам нужны синхронизация и резервное копирование. Вы не хотите, чтобы новые браузерные приложения использовали ваши данные — вместо этого вы хотите только перемещать данные из одного места в другое. Вам нужна синхронизация рабочих компьютеров и стороннее резервное копирование в доме друга. И, наконец, вы не хотите, чтобы кто-то — включая друга, в доме которого хранятся резервные копии — читал эти файлы.

Статья этого месяца посвящена вопросам разработки, на примере *Syncthing*, *EncFS* и *Rsnapshot*. Мы рассмотрим причины принятия определенных решений и предложим инструкции по настройке, распознающие всю архитектуру, включая восстановление файлов. Эксперты безопасности рассмотрели *EncFS*, и нам известны его достоинства и недостатки. *Syncthing* находится в активной разработке с достаточно узким охватом, чтобы он хорошо с этим справлялся. Набор инструментов *Rsnapshot* — вневременной предшественник Apple Time Machine, которой нет равных по эффективности хранения, простоте восстановления потерь и длительности существования на рынке.

Разработка (рис. внизу) использует две одноранговые группы — пиры [peer] *Syncthing*, соединенные каналом шифрования. Как показано в правой части схемы, первая группа пиров является накопителем файлов с ваших рабочих компьютеров.

СКОРАЯ ПОМОЩЬ

Syncthing сохраняет старые копии файлов, но если вам нужно решение для перемещающегося инкрементного резервного копирования, *rsnapshot* может сделать это для вас, и он прекрасно согласуется с *Syncthing*.



Накапливайте, шифруйте и делайте резервные копии своих файлов дистанционно с помощью методов, описанных в этой статье.

Обратный режим	Encfs над Sync	Те же директории Root	Encfs под sync
Открытый текст (реальные данные) Sync Ring	Sync работает без EncFS	Sync работает без EncFS	Sync работает без EncFS
Шифрованный (виртуальные данные) Sync Ring	Sync «папка отсутствует» когда EncFS останавливается	Sync «Остановлен» когда EncFS работает	Удаление файлов, если EncFS останавливается

EncFS по-разному взаимодействует с *Syncthing*, в зависимости от взаимоотношений двух директорий root. Использование EncFS в реверсивном режиме не разрешает взаимоотношения, которые мы хотим использовать.

Она синхронизирует файлы между локальными рабочими компьютерами, причем за любой из них может сесть некий пользователь и начать работу с данными.

Мостом может служить любой компьютер; однако особенно хорошо подходят для этой задачи компьютер Raspberry Pi 3 и операционная система Raspbian Linux. Raspberry Pi 3 имеет беспроводное соединение 802.11 и может хранить терабайты данных с помощью дисков USB. Нам не нужен быстрый компьютер, потому что файлы данных не используются в реальном времени, как решение для хранилища с сетевым доступом. Это маломощное устройство, и мы можем оставить его включенным всё время, подсоединив его к универсальному источнику питания.

Мы использовали Pi в качестве устройства без монитора, мыши и клавиатуры, которое загружает VNC, SSH, EncFS и *Syncthing* во время загрузки, когда *cron* запускает *Rsnapshot*. При необходимости мы можем получить доступ к нему с помощью GUI *UltraVNC* или SSH командной строки с одного из наших рабочих компьютеров.

Хотя нам вполне подходит компьютер Pi, рекомендации здесь могут относиться к любому компьютеру с Linux. В остальной части нашей статьи мы будем именовать компьютер Pi «мостом».

Сбор данных воедино

Для автоматизации процесса накопления данных с помощью *Syncthing* имеется три метода, которые можно использовать для накопления данных с рабочего компьютера на мосте:

1 Назовите корневые синхронизируемые директории [sync root] на рабочем компьютере и используйте *Syncthing* для исключения подфайлов или поддиректорий, которые вы не хотите собирать. Номинально это может быть вашей домашней директорией на каждом рабочем компьютере, если рабочие компьютеры имеют одну и ту же структуру домашней директории. Считайте имеющуюся корневую директорию «шиной данных», с помощью которой каждый узел может выбрать, на что не подписываться или чему не содействовать. В данном случае sync root будет тем же, что и EncFS на мосте.

2 Вызовите несколько директорий sync root на рабочих станциях и поместите каждую как пир в синхронизируемую корневую директорию моста. Номинально эти директории-пиры могут быть документами, фотографиями, музыкой и т.д. Это, похоже, неплохо работает, когда у рабочих станций разные операционные системы и иерархии имен путей интересующих вас данных. В данном случае EncFS root будет выше отдельных синхронизируемых директорий.

3 Создайте резервную проходную директорию на каждом рабочем компьютере и создайте символические ссылки в этой директории на любую локальную директорию или локальный файл на рабочей станции. Затем синхронизируйте проходные директории с помощью *Syncthing*. Это обеспечит максимальную гибкость на каждой рабочей станции, чтобы переносить или не переносить определенные файлы или директории в группу с распределенным доступом. Sync root будет такой же, как EncFS root на мосте.

В *Syncthing* имеется флаг “send only [только отправка]”, который можно присвоить любому узлу. Однако в нем не имеется назначения “receive only [только получение]”. Иными словами, нельзя установить флаг на мосте, и поэтому изменения не будут передаваться назад на рабочие станции — в случае ошибки, вы же не хотите, чтобы вред был нанесен всем вашим рабочим

станциям! На первый взгляд каждая рабочая станция может использовать назначение “send only”, чтобы защитить себя, но это бы обходило стороной законную потребность получать изменения на одноранговые рабочие компьютеры.

Если вас беспокоит угроза того, что с моста будут передаваться изменения, есть еще один способ обеспечить поток данных только с рабочих компьютеров к мосту — вам следует обратить внимание на такие решения, как *Midnight Commander*, *Rsync* или *FileZilla*. Они могут работать через SSH-соединение. *Rsync* — утилита командной строки, которая копирует файлы с рабочих компьютеров на мост. Вы можете получить одностороннюю синхронизацию в полуреальном времени, используя *Rsync* в качестве плановой процедуры *cron*. *FileZilla* предлагает двухпанельный GUI, дающий возможность передачи вручную в любом направлении. Мы также использовали *Midnight Commander* на Linux или подсистему Windows Linux или *puTTY* на Windows 10. Вы также можете использовать обычный сетевой протокол распределенного доступа к файлам своей встроенной операционной системы.

Шифрование

EncFS работает только на мосте и соединяет два кольца синхронизации [sync ring]. Основанные на Linux версии encfs предлагают два способа хранения данных на мосте.

1 Храните незашифрованные данные и шифруйте их «на ходу» в виртуальной точке монтирования, когда ее нужно будет синхронизировать. EncFS называет «обратный» режим [“reverse” mode]. Это оказывается самым простым способом, потому что исходные данные не затрагиваются. Однако в нем есть проблемы.

2 Храните данные зашифрованными и расшифровывайте их «на ходу» в виртуальной точке монтирования, когда это нужно для локальных операций. EncFS называет это «прямым» режимом

СКОРАЯ ПОМОЩЬ

Не ограничивайтесь только одним внешним размещением. Если у вас много друзей или родственников, то у вас могут храниться зашифрованные резервные копии, синхронизированные с каждым из них, для большей надежности и эффективности.

» ПЛЮСЫ ЭТОЙ АРХИТЕКТУРЫ

- Вы владеете правами на данные, и несете ответственность за данные и администрирование. Только вы и доверенные внешние партнеры обладают вашими данными, но исключительно вы можете их расшифровать.
- Вы не платите за работу или подписку.
- Синхронизация проводится без сторонних реле. Всё, что вам нужно — найти друга со свободным местом на диске и подключением к Интернету.
- Резервное копирование происходит на основе определенных событий (изменение файла) и времени (весьма часто).
- Внешние резервные копии масштабируются согласно вашим потребностям.
- Данные шифруются вне вашей локальной сети и в состоянии покоя.
- Архитектура подходит для удаленно хранящихся данных, а не для произвольного доступа в реальном времени к удаленным данным.
- Никаких навязанных просмотрщиков файлов/слайд-шоу или музыкальных программ, быстро устаревающих.
- Никаких посторонних баз данных или требований по компьютерным языкам, например, установить MySQL, PHP, web-серверы и т.д.
- Идеальная для архитектуры цепи инструментов Linux: независима от данных, инструменты за сценой, тихая работа.
- Каждый рабочий компьютер или устройство хранит свои данные в местоположении по умолчанию — например, MyDocs, Home, Documents — тогда как мост накапливает данные и выполняет резервное копирование и синхронизацию.



СКОРАЯ ПОМОЩЬ

Syncthing не ограничивается другими десктопами или ноутбуками. Одним из ваших рабочих компьютеров или одной из ваших внешних резервных копий может быть смартфон или планшет.

[“forward” mode]. Именно так работают большинство пакетов шифрования, ради дополнительной безопасности, необходимого для хранения данных зашифрованными — даже на вашем локальном диске.

Между *Syncthing* и *EncFS* существует взаимодействие, которое надо учесть и которое не очень обсуждается на web-форумах или в программной документации. Это уникально для версий *EncFS*, которые разрешают и прямой, и обратный режимы, например, для версии Linux.

Когда *EncFS* не работает, его директория виртуальной точки монтирования оказывается пустой (или отображает всё то, что у вас есть по этому пути). Представьте себе ситуацию, когда *Syncthing* синхронизирует виртуальную директорию, а *EncFS* еще не работает или уже остановился. В этой ситуации *Syncthing* распространит пустоту точки монтирования на всю одноранговую группу, а вас, вероятно, такое не очень обрадует!

Чтобы избежать проблем с пустой точкой монтирования, *Syncthing* создает файл с нулем байт, **.stfolder**, в каждой директории sync root и использует присутствие этого файла как указание на то, что монтирование работает как положено. Мы будем использовать этот семафорный файл в качестве указания на то, что *EncFS* работает, что, в свою очередь, контролирует, будет *Syncthing* синхронизировать или нет.

» ОБРАТНЫЙ РЕЖИМ ENCF S

Обратный режим *EncFS* хранит открытые текстовые файлы и использует виртуальную точку монтирования для представления зашифрованной передачи файлов.

На стороне открытого текста (см. таблицу на стр. 67) *Syncthing* создает **.stfolder** в директории открытого текста, где хранятся реальные данные, поэтому одноранговая группа открытого текста *Syncthing* на мосте всегда будет работать, вне зависимости от того, представляет ли *EncFS* зашифрованные файлы на другой стороне. Это хорошо, независимо от того, являются ли одинаковыми sync root и encfs root, или один вложен в другой.

Для виртуальной директории зашифрованной стороны, если sync root ниже *EncFS* root и *EncFS* останавливается, то синхронизируемая папка исчезнет, и вы увидите в *Syncthing* ошибку «отсутствует папка [folder missing]». Это не проблема, потому что *Syncthing* возобновит работу, когда возобновит работу *EncFS*.

Однако с разными взаимоотношениями между директориями root имеются проблемы. Если у *EncFS* и *Syncthing* одна и та же директория root, то когда *EncFS* запускается, он занимает точку монтирования и в точке монтирования делает видимыми только зашифрованные версии файлов. *Syncthing* не увидит никакого **.stfolder** в нижележащем монтировании и введет статус «остановлен [Stopped]», что нежелательно. Если вы попытаетесь создать **.stfolder** после запуска *EncFS*, вы получите ошибку создания файла “Input/Output”, поскольку **.stfolder** не может быть незашифрованным.

Если sync root выше *EncFS* root, то **.stfolder** всегда будет присутствовать в sync root, независимо от того, делает ли *EncFS* все зашифрованные файлы видимыми в поддиректории. Если внешняя одноранговая группа *Syncthing* синхронизируется, пока *EncFS* не работает, отсутствие зашифрованных файлов будет распространяться по всем вашим внешним местам резервного копирования (читай: удалятся все файлы), а вам это вряд ли нужно!

Образцы такого нежелательного поведения см. в таблице вверху стр. 67.



Мост Raspberry Pi, круглосуточно работающий в свободной комнате. Доступен из любой точки мира!

Резервное копирование данных

Синхронизация — это не то же самое, что стратегия резервного копирования. Если в одном месте что-то произойдет — например, случайное удаление файла — синхронизация может распространить эту ошибку на все компьютеры. Синхронизация — это вспомогательная функция, дающая возможность доступа к своим файлам с разных устройств, не беспокоясь об обновлении данных до свежих версий вручную. Обновления утомительны и подвержены ошибкам, тогда как автоматическая синхронизация избегает обеих этих проблем.

Синхронизация строится на признании того, что основная ценность и цель вашего компьютера (компьютеров) не в операционной системе или приложениях и программах или определенных устройствах, которые вы используете. Наоборот, всё это — инструменты доступа к вашим данным; сами данные и являются главной ценностью.

Помимо синхронизации, нам нужна возможность резервного копирования ваших данных. Есть как минимум три причины делать резервные копии.

Первая — резервные копии защищают вас от сбоев оборудования, например, отказа жесткого диска. Для защиты от этой угрозы будет достаточно сделать копию на отдельном диске, присоединенном к тому же самому компьютеру.

Второй тип резервного копирования ведет исторический архив для разных моментов времени. Он отвечает необходимости «отката» к старым версиям в случае, если что-то будет повреждено или потеряно. Вы не знаете, какая старая версия будет наилучшей, поэтому стоит иметь несколько. Примерами подобного резервного копирования являются Apple Time Machine или *Rsnapshot* на платформах Linux.

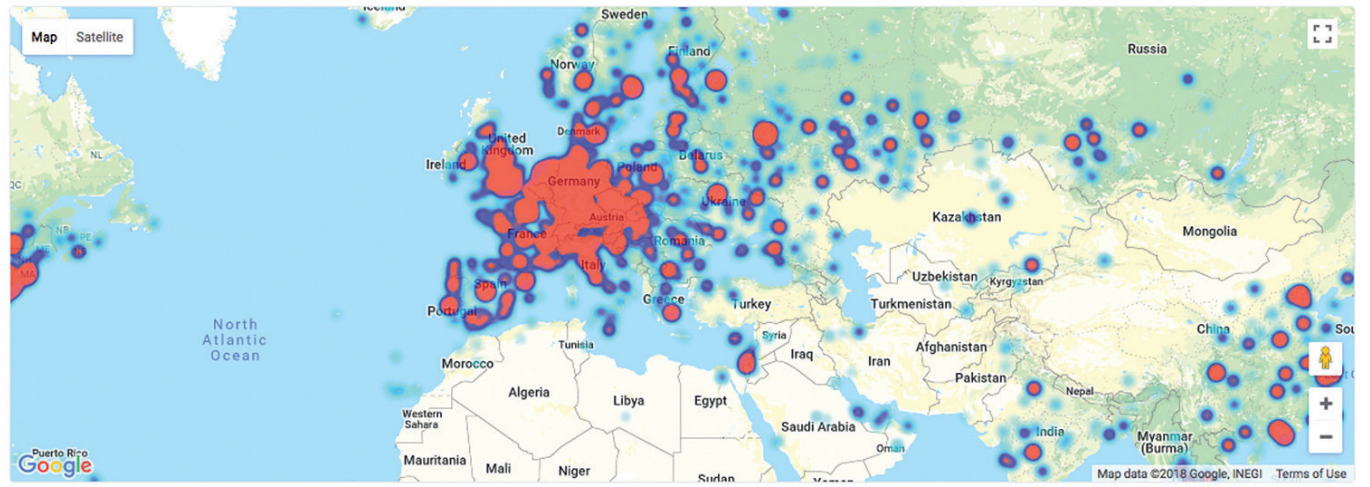
Третий тип резервного копирования — внешнее резервное копирование, которое поддерживает копию ваших данных в другом месте. Это защищает от риска кражи или пожара или иных подобных случаев, которые могут целиком опустошить их пристанище. Резервное копирование по второму типу автор уже выполняет, так что с новой разработкой мы реализуем именно третий тип.

Прямой режим	Encfs выше sync	Одинаковые директории Root	Encfs ниже sync
Открытый текст (виртуальные данные) Sync Ring	Sync “folder missing” в остановленном EncFS	Sync “Stopped” если EncFS остановлен	Удаляет файлы, если EncFS остановлен
Зашифрованный (реальные данные) Sync Ring	Sync работает без EncFS	Sync работает без EncFS	Sync работает без EncFS

При использовании *EncFS* в прямом режиме взаимодействие между *EncFS* и *Syncthing* не позволяет реализоваться только одному отношению между директориями root

Usage Metrics

This is the aggregated usage report data for the last 24 hours. Data based on **32522** devices that have reported in.



Хотя авторы *Syncthing* утверждают, что их программа не предназначена для резервного копирования, она умеет выполнять и эту функцию. В нашем сценарии компьютер-мост имеет большой диск USB, который служит репозиторием *Rsnapshot*. В диаграмме архитектуры он показан наполовину в круге моста и наполовину за пределами этого круга, поскольку это может быть локальный диск или отдельная внешняя резервная копия.

Ключевое различие между *Rsnapshot* и *Syncthing* в том, что *Rsnapshot* является резервным копированием по расписанию (каждый час, например), тогда как *Syncthing* обновляет удаленные компьютеры при каждом изменении файла.

Кроме того, перенос архивов *Rsnapshot* на удаленный компьютер архитектурно сложно. *Rsnapshot* ограничен «вытягиванием» резервных копий и требует нестандартных разрешений файлов для обеспечения безопасности на многопользовательской системе. Это можно обойти, смонтировав удаленный ресурс моста на вашей локальной рабочей системе. В качестве альтернативы запустите *Rsnapshot* на удаленном сервере и перетащите *rsync*, через SSH, или протолкните на удаленный сервер, и затем пусть *Rsnapshot* на удаленном сервере работает инкрементно.

На нашем уроке мы не охватываем установки *Rsnapshot*, потому что она модульная и отлично документирована. Если вам это нужно, добавьте ее параллельно всему остальному, описанному здесь, и она никак не повлияет на работу *Syncthing/EncFS*. Если в своей группе *synс* вы используете Mac, можете прибегнуть к функции Time Machine.

Внешняя синхронизация данных

Одноранговая группа *Syncthing* в левой стороне схемы перемещает данные внешне, отправляя зашифрованную версию данных в один конец на внешние сайты резервного копирования. Мы решили использовать *Syncthing* по четырем причинам:

- 1 Сетевой GUI *Syncthing* подходит конечным пользователям на разных платформах.
- 2 Данные обновляются при изменении файла, а не по расписанию обновлений, как в *Rsnapshot*.
- 3 *Syncthing* уже используется для синхронизации внутренних рабочих компьютеров.
- 4 Легко настроить многочисленные внешние устройства, включая смартфоны, что смягчает последствия отпадения какого-либо места, обеспечивая почти 100% безотказной работы.

Syncthing позволяет при желании использовать более одного удаленного компьютера, причем вся дополнительная настройка сводится к тому, чтобы сообщить удаленному оператору

соответствующий Key ID. Любой удаленный оператор может попытаться войти в ваше кольцо синхронизации, введя Key ID. Однако рабочая станция, которая создала Key ID, должна однократно авторизовать подключение с помощью панели управления *Syncthing*.

Учтите, что *Syncthing* синхронизируется только с отдельным устройством. Режим “Send and Receive [Отправка и Получение]” подходит для удаленных компьютеров, чтобы они могли помочь друг другу в обновлениях. Обратите также внимание, что *Syncthing* на мосте в левой части схемы имеет назначение “Send Only [Только Отправка]”, так что удаленные компьютеры не могут отправить изменения назад — случайно или намеренно. Если в какое-то время на внешних устройствах созданы сомнительные файлы, вы можете довольно легко заставить произвести перераспространение «истинного набора» данных с помощью *Syncthing* GUI моста.

Закончив работу с архитектурой, в статье в следующем месяце мы расскажем, как настроить определенные программы. Желая с этим разобраться, скачайте себе *Syncthing* и *Encfs*, да подготовьте их к работе! **LXF**

Оptionальные анонимные отчеты *Syncthing* на <https://data.syncthing.net> весьма обширны и прямо-таки волнуют.

» ВЫБОР ПРЯМОГО РЕЖИМА ENCFs

Прямой режим *EncFS* хранит зашифрованные файлы и использует виртуальную точку монтирования для открытых текстовых перемещений файлов.

На стороне открытого текста изначально запустите *Syncthing* для создания **.stfolder** в директориях открытого текста во время работы *EncFS*, чтобы **.stfolder** распространился на зашифрованный файл в левой стороне схемы. С одинаковыми *sync root* и *EncFS* файл **.stfolder** снова появится на стороне открытого текста в любое время, когда работает *EncFS*, но его там не будет, если *EncFS* не работает. Именно это нам и нужно, и *Syncthing* сообщит о статусе “Stopped [Остановлен]”, если файла там нет. С *EncFS* выше *sync root* *Syncthing* выдает сообщение “folder missing [Отсутствует папка]”, если *EncFS* не работает. Не помещайте *EncFS* root ниже *sync root*, потому что если *EncFS* не работает, *Syncthing* всё равно синхронизирует пустую директорию назад на рабочие компьютеры и удалит файлы.

На зашифрованной стороне открытый текст **.stfolder** размещается *Syncthing* при изначальном запуске. Если *sync root* и *EncFS* root одинаковы, *EncFS* просто оставляет его в покое, и он всегда будет присутствовать. Если *EncFS* остановится, синхронизация будет излишней, но не вредной. Если *sync root* ниже *EncFS* root, синхронизация запустится, даже если *EncFS* не работает.

Это поведение отражено в таблице на стр. 68.

NODE-RED

Автоматизируем жизнь сетевой «вещью»

Тим Армстронг помогает вам освоить эфир с помощью беспроводной «вещи» в интернете, используя LoRa, MicroPython и TheThingsNetwork.



НАШ
ЭКСПЕРТ

Тим Армстронг работает сетевым и системным архитектором в Nerdalize. Он специализируется на разработке и внедрении облачных сред и ISP-сетей.

На этом уроке мы возьмем реальные данные, собранные в TheThingsNetwork, и представим их на простом сайте. Для создания нашего сайта мы используем Node-RED, это отличный язык программирования на основе графических блок-схем. Вы перетаскиваете узлы, представляющие входы, функции и выходы, в блок-схему, которую можно развернуть как службу на чем угодно, от Raspberry Pi до облачной платформы.

Ранее в **LXF235** мы собрали простой датчик, который измерял напряжение на светочувствительном резисторе (LDR) и посылал эти данные через LoRa в TheThingsNetwork (TTN). Эта схема применима к большинству аналоговых датчиков (и даже некоторым микрофонам) — простой заменой LDR на желаемый аналоговый датчик.

Node-RED предлагается как «шаблонное приложение» на IBM Bluemix. Мы можем просто зарегистрировать бесплатную учетную запись разработчика (называемую учетной записью lite [облегченной]) и запустить пример. Это позволяет нам начать работу над тем, что мы хотим сделать, не задумываясь о развертывании, установке, управлении и обновлении программного обеспечения сервера. Всё это сделает Network Operations Center от IBM.

Node-RED на Bluemix

Чтобы получить учетную запись lite на Bluemix, зайдите на <https://console Bluemix.net/registration>. Заполните форму и нажмите Create Account. Затем надо будет открыть ссылку в своей почте, чтобы подтвердить регистрацию. Вы можете просто пропустить вводный обзор, потому что по сути он мало что дает. По умолчанию вы автоматически получаете учетную запись lite, и мы можем перейти прямо в каталог и выбрать Node-RED Starter из раздела **Boilerplates** в каталоге.

Дайте своему приложению уникальное имя, выберите регион для развертывания и нажмите Create. После запуска можно перейти к блочному редактору Node-RED, щелкнув ссылку Visit App URL рядом с названием нашего приложения, и следуйте указаниям мастера установки, чтобы защитить свой экземпляр Node-RED. Наконец, нажмите ссылку Go на ваш блочный редактор Node-RED.

Итак, мы находимся в блочном редакторе и можем видеть палитру узлов слева, панель отладки справа и рабочую область посередине. Теперь нам надо добавить пару наборов узлов в нашу палитру. Поскольку мы создаем сайт панели мониторинга, где мы можем видеть и взаимодействовать с данными нашего устройства IoT, мы добавим набор node-red-dashboard, а поскольку наше устройство IoT передает данные в TTN, мы добавим набор node-red-contrib-ttn. Для этого надо изменить установку и переустановить наш экземпляр Node-RED через CLI Cloud Foundry.

Чтобы установить CLI CloudFoundry, мы можем загрузить его по ссылке на странице Getting started раздела Node-RED нашей консоли Bluemix. После его загрузки откройте терминал в той же папке и выполните следующее:

```
$ sudo dpkg -i cf-cli-installer_6.34.1_86_64.deb
```

Пока идет установка, загрузите стартовый код нашего развертывания с консоли Bluemix, а затем распакуйте его. Затем откройте файл **package.json** в любимом текстовом редакторе и вставьте следующие строки наверх раздела зависимостей:

```
"node-red-contrib-ttn": "2.0.x",
"node-red-dashboard": "2.8.x",
```

Сохраните изменения, и следующий шаг — открыть терминал в каталоге стартового кода и установить URL Cloud Foundry API

IBM Bluemix похожа на коллекцию меньших, чем одна большая, облачных платформ со всеми сервисами: от SoftLayer Bare-Metal до Watson Natural-Language.

в Bluemix API для региона, в котором вы решили развернуть свой экземпляр Node-Red:

```
$ cf api https://api.<REGION>.bluemix.net
```

Войдите при помощи

```
$ cf login -u <YOUR_EMAIL_ADDRESS>
```

Теперь, войдя в систему, мы можем передать нашу измененную установку на Bluemix, используя

```
$ cf push <YOUR_APP_NAME>
```

По завершении повторного развертывания надо будет обновить страницу блочного редактора и войти в систему, используя учетные данные, созданные ранее в мастере настройки Node-RED.

Входящие данные

Распространенным заблуждением является то, что TTN хранит данные. Видимо, это связано с тем, что у него есть удобная вкладка Data в консоли. Но ни TTN, ни Node-RED данные не хранят, и первое, что нам надо сделать — это настроить наше хранилище данных, чтобы мы могли хранить входящие данные.

Поскольку Node-RED хранит свою конфигурацию и блоки в CouchDB на Bluemix под названием Cloudant, можно использовать тот же экземпляр cloudant и держать наши данные там.

Для этого перетащите узел cloudant out из раздела storage нашей палитры и дважды щелкните по нему. Появится панель настройки. На панели настройки мы можем выбрать наш сервис cloudant, затем дать нашей новой базе данных имя — давайте назовем ее lory. Позаботьтесь установить для нашей операции значение insert и флажок Only store msg.payload object. Затем можно нажать кнопку Done в правом верхнем углу панели настроек.

Далее нам нужен источник данных, который в этом случае является блоком ttn uplink и находится в разделе Inputs нашей палитры. Опять же, надо его настроить, так что дважды щелкните по нему, чтобы открыть панель. В разделе App вы заметите, что справа имеется значок карандаша. Нажатие на него вызовет меню TTN access key. Чтобы его заполнить, нам нужны несколько полей из нашей консоли TTN (<https://console.thethingsnetwork.org>), поэтому давайте откроем ее в другой вкладке.

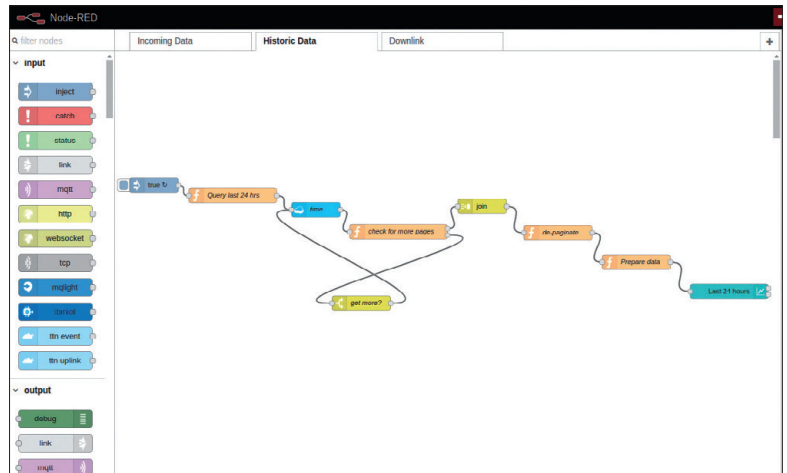
В консоли TTN перейдите в Applications и выберите наше приложение, а в разделе Access Keys внизу страницы нажмите Manage keys и затем Generate new access key. Дайте новому ключу имя и убедитесь, что отмечен только один флажок — Messages. Затем нажмите кнопку Generate Access Key в правом нижнем углу и вернитесь на вкладку Overview. В разделе Access Keys вы увидите наш новый ключ. Скопируйте его в буфер обмена, щелкнув по значку буфера обмена справа от затемненного ключа.

Вернувшись на вкладку Node-RED, мы можем ввести наш Application ID и вставить Access Key в соответствующие поля. Наконец, нажмите Add, а затем Done, чтобы сохранить настройки.

Ну вот, у нас есть наш вход и наше хранилище готово; надо только добавить отметку времени для входящих данных и сохранить их. Поскольку это не имеющийся блок, который мы просто вставили, надо использовать блок функции, чтобы сделать это самим. Блок функции позволяет изменять содержимое объекта msg с помощью JavaScript. Соглашение предписывает, чтобы функциональный блок выполнял только одну задачу, четко обозначенную его именем. Для этого перетащите блок функции в схему, дважды щелкните по нему, назовите его Add Timestamp и добавьте следующую строку в начало:

```
msg.payload.time = Date.now();
```

Нажмите Done, и теперь пора соединить блоки в нашей первой схеме. Нажмите и перетащите из точки справа от блока TTN uplink в левую точку функционального блока, затем сделайте то же самое между блоками Add Timestamp и cloudant. Наконец, нажмите кнопку Deploy в правом верхнем углу консоли. Теперь каждое измерение от нашего LoPy, полученное шлюзом LoRa



Node-RED действительно удобен для быстрого создания web-приложений и чат-ботов.

и отправленное в The Things Network, будет снабжено отметкой времени, а затем сохранено как документ в нашей базе данных.

Увидеть значит поверить

Хранение данных полезно, но в таком случае надо извлекать их из базы данных, прежде чем мы сможем их увидеть — и мы еще дойдем до этого; но пока мы имеем дело с «живыми» данными, давайте создадим наш первый элемент панели мониторинга. Для этого нам нужен еще один функциональный блок и измерительный блок: его вы можете найти в разделе dashboard палитры. Наша функция в этом случае может быть названа get_volts и выглядит просто:

```
msg.payload = msg.payload.millivolts/1000; return msg;
```

Затем проведем еще одну линию из правой точки нашего блока TTN uplink, на сей раз подключив ее к нашей функции get_volts, после чего протянем линию от ее правой точки к измерительному блоку. Теперь мы снова можем нажать кнопку Deploy, а затем на вкладке dashboard нашей панели отладки щелкнуть маленький всплывающий значок в правом верхнем углу, который должен открыть новую вкладку, показывающую наш измеритель, и через

СКОРАЯ ПОМОЩЬ

Учетные записи lite на Bluemix хороши тем, что пока вы только разрабатываете, вы ничего не платите, а когда вы готовы к развертыванию, вам остается всего пара щелчков до полной мощи Bluemix Cloud от IBM.

» ИСХОДЯЩИЙ КАНАЛ TTN

До сих пор мы рассматривали восходящий канал от нашего устройства к облаку, но TTN также обеспечивает удобный способ отправки нисходящих сообщений на устройство, которое может быть использовано для запуска действия на устройстве. Для этого начните новую схему и перетащите блок "ttn downlink", дайте ему имя, задайте номер порта равным 1, установите флажок подтверждения, укажите идентификатор устройства и установите расписание для последнего.

Потом мы можем просто отправить этому блоку массив байтов как msg.payload, и он будет получен в очередь приема LoRaMQ.

Например, можно добавить блок переключателя панели мониторинга и подключить его к функциональному блоку, который будет отправлять массив байтов на устройство, когда переключатель изменит состояние:

```
payload = new Uint8Array(2);
payload[0]=9;
payload[1]=msg.payload;
msg.payload = payload;
return msg;
```

Затем, подключение выхода этого блока ко входу блока ttn downlink и нажатие deploy означало бы, что каждый раз, когда вы изменяете переключатель на панели управления, устройство получает сообщение, содержащее этот массив байтов, который затем может быть декодирован и запущено действие на устройстве.

СКОРАЯ ПОМОЩЬ

Называйте все свои узлы подходящими именами. Это не только сэкономит вам массу времени, когда вы вернетесь к коду позже, но вы еще сможете использовать панель отладки для пометки ошибок; так будет проще отлаживать схему.

короткое время она должна показать последнее полученное значение от нашего LoPy.

Сидя на работе

Мы обзавелись первым работающим элементом панели мониторинга, и пора подумать о втором. Вернемся к редактору Node-RED и начнем следующую схему. Для этого надо щелкнуть по маленькой вкладке [+] в правом верхнем углу нашего рабочего пространства. Получение истории немного сложнее, поэтому мы создадим ее в разделах, в частности, Query, Response и View.

Чтобы начать нашу схему, нам нужен вход; в данном случае мы используем блок inject, поэтому перетащите его в схему и дважды щелкните по нему. Затем установите его содержание как "boolean", повторение с интервалом, и установите на запуск каждые 120 секунд.

Вследствие принципа работы Cloudant нам надо создать индекс поиска, чтобы мы могли позже его запросить. В данном случае это поле "time", которое мы ввели в первую схему. Для этого надо вернуться к нашей консоли Bluemix (<https://console.bluemix.net/dashboard/apps/>) и выбрать экземпляр Cloudant, который для нас был автоматически создан, а затем нажать кнопку Launch. Теперь нам должны быть представлены две наши базы данных; дважды щелкните по той, что называется lopy.

Чтобы создать наш индекс поиска, щелкните (+) рядом с пунктом Design Documents в меню слева. Затем в подменю выберите New Search Index: давайте назовем его by_timestamp, а наш индекс —time. Затем в разделе функций замените index("name", doc.name); на index("time", doc.time); и нажмите кнопку Create Document and Build Index внизу страницы. Чтобы проверить наш индекс, скопируйте time: [0 TO Date.now()] в поле запроса и нажмите Enter. Это должно вернуть все данные наших измерений.

Теперь мы можем вернуться к Node-RED и завершить наш раздел Query. Для этого нужен еще один функциональный блок и блок cloudant in, поэтому давайте перетащим их в схему и дважды щелкнем по блоку функции. В этой функции надо заменить содержимое запросом, подобным тому, что мы использовали для тестирования нашей базы данных, но поскольку на сей раз нас интересуют только последние 24 часа, вставьте msg.payload="time:[\"+(Date.now()-86400000)\" TO \"+Date.now()+\"]"; и перевод строки.

Переходя к блоку cloudant, нам надо настроить его на использование только что созданного индекса поиска. Задайте имя базы данных lopy, затем установите поиск по Search index. Слева в следующих двух полях введите by_timestamp, а справа — time; затем нажмите Done.



Облачно, возможны осадки – это не о том, чем является The Things Network...

Наконец, подключите блок inject к левой точке функции запроса, а выход функции — к блоку cloudant. Если вы хотите протестировать это сейчас, перетащите блок debug, подключите его к выходу cloudant и нажмите Deploy. Затем при нажатии квадрата слева от блока Inject вы увидите данные на панели отладки в виде массива до 200 результатов.

Размер имеет значение

Как вы могли заметить, 24 часа, разделенные на 100-секундные выборки, составят 864 результата. Это не максимум 200, которые дает нам блок cloudant; а как же получить прочие 664 результата? Это приводит нас к разделу Response. Если установить параметр Output блока debug для завершения объекта msg, нажать Deploy, а затем запустить блок inject, то вы увидите раздел cloudant для нашего объекта. Развернув его, вы увидите, что он содержит элемент под названием bookmark. Закладка указывает на следующую страницу в ответе на наш запрос. Для его использования нам понадобятся блоки function, switch и join.

Блок join удерживает каждое полученное сообщение до тех пор, пока не выполнится заданное условие, а затем отправляет пакет далее единым целым. В нашем случае этим условием будет, если наш объект msg содержит свойство complete, и мы хотим, чтобы он объединил наши объекты msg в массив. Чтобы установить это, откройте настройки join, затем установите его режим в ручной, его "combine each" — в msg.payload и "to create" — в "an Array".

Блок switch передает любое полученное сообщение, если оно соответствует некому простому критерию. Мы хотим проверить, что msg.payload не навьется нулю — "is not null". Чтобы включить этот блок, измените раскрывающийся список, который сейчас установлен в "=", на "is not null".

Затем надо настроить функцию, в которой мы должны проверить, имеет ли содержимое, получаемое от блока cloudant, какие-либо результаты. Если результаты там есть, мы хотим передать данные на join и перейти на следующую страницу. Если он пуст, то мы получили все страницы и можем инициировать объединение, установив свойство msg.complete в сообщении в join и отправив нулевое содержание переключателю. Для этого надо использовать два выхода: один для нашего блока join и один для нашего блока switch. Вы можете контролировать количество выходов непосредственно под разделом кода. Когда у нас есть несколько выходов, в возврате надо дать список с тем же количеством объектов msg, что и количество выходов.

```
if (msg.payload.length > 0){
  next_page = {payload: {}}
  next_page.payload.query="time:[\"+(Date.now()-86400000)\" TO \"+Date.now()+\"]";
  next_page.payload.bookmark=msg.cloudant.bookmark;
  return [msg, next_page];
} else {
  msg.complete = true;
```

Name	Size	# of Docs	Actions
lopy	1.2 MB	5038	[+], [lock], [trash]
nodered	111.8 KB	4	[+], [lock], [trash]

Экземпляр CouchDB, размещенный на IBM Bluemix Cloudant.


```
return [msg, {payload:null}];
}
```

Теперь подключите выход нашего блока cloudant ко входу функции, затем верхний правый выход нашей функции ко входу нашего блока join, нижний выход функции — ко входу switch и выход нашего switch ко входу нашего блока cloudant.

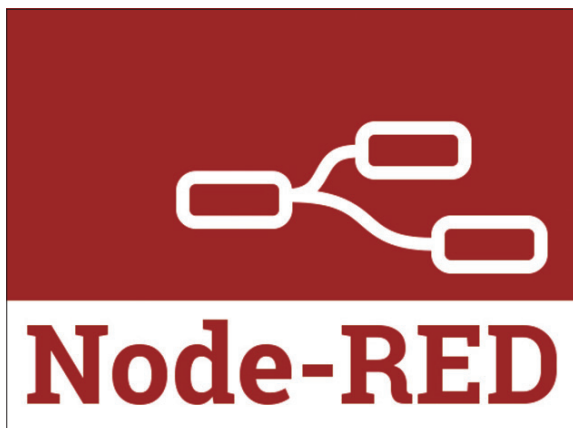
Гарний вигляд!

Наконец, надо предварительно обработать наши данные, чтобы их можно было отобразить на нашем 24-часовом графике. Из-за способа работы блока join у нас есть данные, содержащие массив данных с каждой страницы, полученной из базы данных. Таким образом, первым этапом нашей предварительной обработки будет считывание данных с каждой страницы и привязка содержимого каждой страницы к одной, которую мы отсортируем по метке времени. Для этого перетащите новый блок функции и скопируйте следующее:

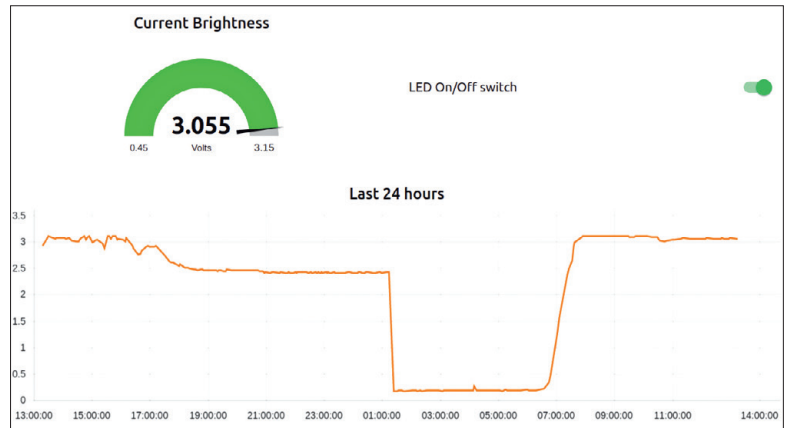
```
new_msg = {}
payload = []
for (i=0; i<msg.payload.length; i++){
  if (msg.payload[i].length > 0){
    payload = payload.concat(msg.payload[i])
  }
}
payload.sort(function(a,b){return b.time-a.time;})
new_msg.payload = payload
return new_msg;
```

Графику из палитры dashboard требуется, чтобы данные были представлены в виде массива, содержащего один объект, имеющий два свойства: одно называется серией, содержащей массив меток, а другое называется данными, содержащих массив массивов, которые содержат объекты с характеристиками x и y. Озадачены? Не волнуйтесь, это не самый очевидный формат данных для тех, кто раньше не работал с библиотеками графиков. С основной целью создайте новый объект функции и скопируйте в него следующий код:

```
t = Date.now();
payload = {}
payload.series = ["Voltage"];
payload.data = [[]];
for (i=0; i<msg.payload.length; i++){
  row = {x:msg.payload[i].time, y:msg.payload[i].millivolts/1000};
  payload.data[0].push(row);
}
```



Эффект от забавной игры слов несколько сник бы, будь логотип на темно-синем фоне.



Если вы застряли, можете обратиться к блокам и панелям на странице <https://lxf236-diy-iot.eu-gb.mybluemix.net>.

```
return {payload:[payload]};
```

Последнее, но не менее важное: перетащите блок графика из палитры dashboard и установите ее ось X на 24 часа. Подключите выход блока join ко входу функции сбора по страницам, подключите выход функции сбора по страницам ко входу функции форматирования данных и подключите выход функции форматирования данных ко входу графика. Затем нажмите кнопку Deploy, запустите блок inject и перейдите на панель мониторинга. Теперь вы должны увидеть свою готовую панель мониторинга с вашим измерителем текущих данных и с вашим графиком истории данных.

Итак, теперь у нас есть датчик LoRa для сайта. Вы можете добавить датчиков, настроить оповещения или использовать исходящий канал TTN для запуска действия на устройстве с датчиком. Поздравляем с вашей интернет-Вещью! **LXF**

» ЗАПУСК NODE-RED НА RASPBERRY Pi

С IBM Bluemix легко начать работу, но иногда вы просто хотите запускать что-то локально, например, простой проект домашней автоматизации, поэтому Node-RED также работает на Raspberry Pi.

Raspbian максимально упростил работу с Node-RED, предварительно установив его как часть образа. Вы можете найти его работающим на http://<IP-адрес_вашего_RaspberryPi>:1880/.

Чтобы установить дополнительные блоки, которые мы использовали в остальной части этой статьи, просто выполните следующее:

```
$ npm install node-red-contrib-ttn
$ npm install node-red-dashboard
$ node-red-stop
$ node-red-start
```

Если вы хотите следовать этому руководству, используя свой RaspberryPi вместо IBM Bluemix, вам также потребуется установить couchdb, инструкции для чего вы найдете в документации couchdb: <http://docs.couchdb.org/en/2.1.1/install/unix.html#debian-based-systems>.

Затем просто следуйте инструкциям, начиная с раздела, в котором мы перетаскиваем наш первый блок cloudant; но когда вам предложат перейти на консоль Bluemix для интерфейса cloudant, надо будет перейти на интерфейс пользователя couchdb на вашем RaspberryPi. Также есть небольшое изменение насчет настройки блока cloudant для использования CouchDB, но оно задокументировано на вкладке info панели отладки, когда у вас выбран блок cloudant, и здесь мы в это вдаваться не будем.

» ПОДПИШИТЕСЬ НА ЖУРНАЛ LXF FORMAT на www.linuxformat.ru/subscribe/

Машинное обучение: Быстрый старт

Михалис Цукалос делится своим проверенным подходом к обработке больших объемов данных по методам машинного обучения в Python 3.



**НАШ
ЭКСПЕРТ**

Михалис Цукалос — администратор UNIX, программист, администратор баз данных и математик. С ним можно связаться через сайт www.mtsoukalos.eu.

СКОРАЯ ПОМОЩЬ

Хотите больше узнать о машинном обучении? Прочтите книги «Элементы статистического обучения», «Машинное обучение на Python» и «Введение в статистическое обучение с приложениями на R».

В наши дни машинное обучение [machine learning, ML] не только является областью исследования в информатике, но и начинает играть ключевую роль в нашей повседневной жизни. Мало того, в ближайшие несколько лет прогнозируется экспоненциальный рост распространенности машинного обучения. Главное преимущество машинного обучения заключается в том, что оно предлагает новые и уникальные подходы к проблемам, которые мы хотим изучить и решить.

Машинное обучение посвящено извлечению знаний из ваших данных с помощью компьютера — в нашем уроке все примеры будут написаны на Python 3. Однако это очень обширная тема, и она не уместится в рамках одной статьи. И основная цель нашего урока — помочь вам начать работу с различными библиотеками Python 3 и их функциями и узнать, что поддерживает каждая библиотека, и дать вам хороший повод для изучения библиотеки — всех или некоторых.

Итак, без долгих предисловий перейдем к следующему разделу, который представляет собой краткое введение в машинное обучение.

Кратко о машинном обучении

С помощью машинного обучения вы можете обнаружить скрытые шаблоны и информацию, которые иначе трудно было бы распознать. В основе машинного обучения так или иначе лежит статистика, поэтому полезно ознакомиться с некоторыми базовыми определениями, такими как среднее значение, медиана, стандартное отклонение, процентиль и выброс. Три основные области машинного обучения — это контролируемое обучение, неконтролируемое обучение и обучение с подкреплением.

Контролируемое обучение [Supervised Learning] — это определение отношений между некоторыми помеченными переменными. Это переменные, для которых мы знаем, что они представляют, и целевая переменная. Сюда относятся и такие области, как классификация и регрессия. Методы неконтролируемого обучения [Unsupervised Learning] пытаются найти скрытые шаблоны в данных без каких-либо знаний об этих данных, включая их типы. Самая популярная категория неконтролируемого обучения — кластеризация.

Обучение с подкреплением [Reinforcement Learning] позволяет изучить поведение системы на основе полученных результатов, используя такие методы, как методы принятия решений Маркова и методы Монте-Карло.

```
16 print('Test Size:', TESTSIZE)
17
18 iris = datasets.load_iris()
19 X = iris.data[:, [2, 3]]
20 y = iris.target
21
22 print('Labels:', np.unique(y))
23 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
24 sc = StandardScaler()
25 sc.fit(X_train)
26 X_train_std = sc.transform(X_train)
27 X_test_std = sc.transform(X_test)
28
29 ppn = Perceptron(max_iter=40, eta0=0.1, random_state=1)
30 ppn.fit(X_train_std, y_train)
31
32 y_pred = ppn.predict(X_test_std)
33 print('Misclassified samples: %d' % (y_test != y_pred).sum())
34
35 print('Accuracy: %.2f' % accuracy_score(y_test, y_pred))
36 print('Accuracy: %.2f' % ppn.score(X_test_std, y_test))
37
```

На рисунке показан код Python 3 файла `classify.py`, который иллюстрирует использование модуля `scikit-learn` и ирисов Фишера для классификации.

Теперь пора дать некоторые определения. Искусственная нейронная сеть моделирует отношения набора входных сигналов и набора выходных сигналов, по типу работы мозга. Проще говоря, искусственная нейронная сеть с помощью взаимно соединенных узлов решает такие задачи, как обработка сигналов или распознавание образов, применяя машинное обучение. Глубокое обучение — это область машинного обучения, которая занимается очень большими искусственными нейронными сетями.

Обобщенная линейная модель представляет собой статистический метод, который, говоря попросту, прогнозирует поведение данных с помощью моделей линейной регрессии. Метод кластеризации пытается сгруппировать данные в наборах таким образом, чтобы объекты, находящиеся в одной группе, были в некотором смысле похожи. В основном это зависит от типа данных, подлежащих обработке. Наконец, метод классификации (который применяет распознавание образов) использует данные обучения для определения некоторых категорий, а затем помещает новые наблюдения в эти категории.

Примеры с кривыми

Первый пример на нашем уроке будет относительно простым: мы попытаемся найти математическую функцию, которая наилучшим образом соответствует входным точкам данных. Это называется подбором аппроксимирующей кривой, и это один

из самых простых типов машинного обучения. Он тесно связан с математикой.

Код файла **simple.py** выглядит так:

```
#!/usr/bin/env python3
import numpy as np
import warnings
warnings.simplefilter('ignore', np.RankWarning)
points = np.array([(2, 4), (3, 1), (9, 3), (5, 1)])
x = points[:,0]
y = points[:,1]
z = np.polyfit(x, y, 4)
f = np.poly1d(z)
print(f)
```

После запуска файла **simple.py** вы получите следующий результат:

```
$. /simple.py
4      3      2
-0.01075 x + 0.07323 x + 1.009 x - 8.739 x + 17.03
```

Здесь мы пытаемся сопоставить данные с многочленом четвертой степени (полиномиальная аппроксимация), так что файл **simple.py** печатает многочлен с вычисленными коэффициентами. Вычисленная кривая поможет предсказать, как ведет себя небольшой пример данных и чего от него можно ожидать в будущем. Как вы увидите ниже, общая структура программы машинного обучения на Python такова. Сначала вы загружаете требуемые библиотеки Python и импортируете свои данные перед началом обработки. Затем вы начинаете выполнять вычисления, чтобы запустить фазу обработки данных и обучения. Это приведет к созданию модели прогноза для ваших данных, которая часто представляется в виде графического изображения. Иногда может потребоваться перебрать множество разных алгоритмов, пока вы не получите прогностическую модель, которая лучше всего описывает данные. По сути, выбор правильного алгоритма и метода является самой сложной частью машинного обучения.

Классификация

Модуль *scikit-learn* — это библиотека машинного обучения, которая предлагает множество методов машинного обучения. К этим методам относится классификация, которая пытается определить категорию, к которой принадлежит каждый элемент данных.

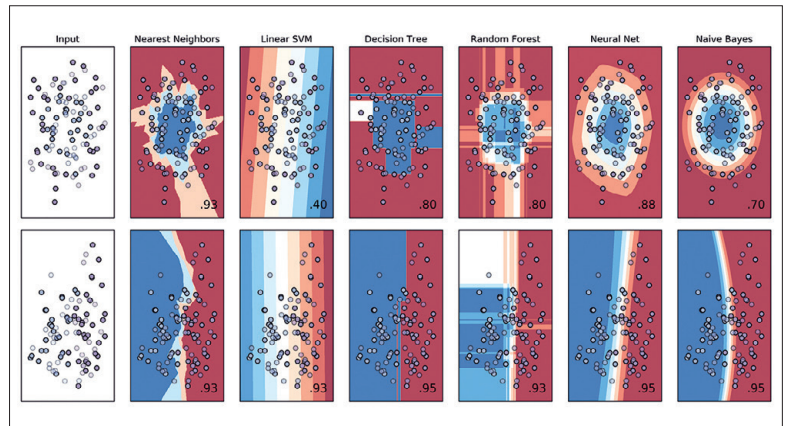
Установить модуль *scikit-learn* проще всего с помощью *pip3*, для этого выполните команду `sudo pip3 install scikit-learn`.

Использование модуля *scikit-learn* показано в скрипте **classify.py**, который приведен на экранном снимке стр. 74. Для простоты в этом скрипте используется набор данных, предоставляемый с модулем *sklearn*, и вычисляются многие свойства набора данных.

Скрипт принимает один аргумент командной строки, который представляет собой процент набора данных, используемых для тестирования — остальные данные будут автоматически использоваться для обучения. Этот аргумент командной строки должен находиться в диапазоне от 0.02 до 0.98.

При запуске файла **classify.py** мы получим следующий результат:

```
$. /classify.py 0.90
Test Size: 0,9
Labels: [0 1 2]
Misclassified samples: 34
Accuracy: 0,75
Accuracy: 0,75
$. /classify.py 0.5
Test Size: 0,5
Labels: [0 1 2]
Misclassified samples: 2
Accuracy: 0,97
```



Это изображение было создано с помощью скрипта **classifyPlot.py** и иллюстрирует классификацию данных с помощью модуля *scikit-learn*.

Accuracy: 0,97

Эти результаты показывают, что использование половины данных для тестирования и половины данных для обучения дает наилучшие результаты с точки зрения точности. Вообще говоря, иметь возможность проверить точность моделей — это всегда хорошо!

Заметьте, что самой сложной задачей является выбор подходящего алгоритма для решения задачи — такой опыт можно получить только практическим путем и учась на своих ошибках. Тем не менее, в скрипте **classifyPlot.py** представлено одно из возможных решений этой проблемы. В нем ваши данные обрабатываются с использованием различных алгоритмов и методов. Большая часть кода в файле **classifyPlot.py** предназначена для построения графика результатов, а большую часть настоящей работы выполняет функция `train_test_split()` из библиотеки *scikit-learn*. Для построения графика используется хорошо известная и мощная библиотека *matplotlib*.

СКОРАЯ ПОМОЩЬ

В языках программирования, включая Go, Python и R, есть пакеты и модули, которые помогут обработать данные с помощью алгоритмов и методов машинного обучения.

» БИБЛИОТЕКИ PYTHON ДЛЯ ML

Хотя вы можете реализовать все алгоритмы машинного обучения почти на любом языке программирования, в Python 3 есть много модулей, которые помогут вам начать работать с машинным обучением без необходимости реализовывать всё с нуля.

Модуль *scikit-learn* создан на основе *NumPy*, *SciPy* и *matplotlib* и предлагает алгоритмы для классификации, регрессии, кластеризации, выбора моделей, уменьшения размерности и предварительной обработки данных. Дополнительные знания о модулях *Numpy*, *matplotlib* или *SciPy* помогут вам, особенно при построении графиков данных. Библиотека *TensorFlow* также идеально подходит для математических операций, таких как операции с многомерными массивами, а также с векторами и матрицами. Существует облегченная версия *TensorFlow* под названием *TensorFlow Lite*, для мобильных и встроенных устройств. Дополнительную информацию о *TensorFlow Lite* можно найти на <https://github.com/tensorflow/tensorflow/tree/master/tensorflow/contrib/lite>.

Еще одна популярная библиотека — библиотека глубокого обучения *Keras Python*, созданная на основе *TensorFlow*, *CNTK* или *Theano*. Дополнительные сведения о *Keras* см. на сайте <https://keras.io>. *CNTK* — это открытый инструмент для глубокого обучения, разработанный Майкрософт. Подробности о нем можно узнать на <https://github.com/Microsoft/cntk>.

Наконец, *Theano* — это библиотека Python для работы с математическими выражениями, содержащими многомерные массивы (*numpy.ndarray*). Кроме того, *Theano* можно использовать для глубокого обучения в Python, поскольку с ее помощью можно создавать модели глубокого обучения. Дополнительные сведения см. на <https://github.com/Theano/Theano>.



СКОРАЯ ПОМОЩЬ

Пропусту говоря, машинное обучение – это изучение прошлого и настоящего для того, чтобы как можно точнее предсказать будущее.

Интересно, что при запуске файла `classifyPlot.py` формируется изображение (показано на рис. вверху стр. 75). Тогда вы с первого взгляда можете понять, что происходит с вашими данными, а также хорошо представить себе результаты для различных алгоритмов классификации. Как видите, при правильном использовании машинное обучение очень полезно и удобно.

Кластеризация

Кластеризация — это способ группировки похожих объектов (по аналогии с людьми, которые толпятся у полок со скидками в супермаркете). В этом разделе будет использоваться алгоритм кластеризации k-средних. Это популярный неконтролируемый алгоритм машинного обучения, способный разделить набор данных на необходимое количество кластеров, которое указывается в качестве аргумента командной строки программы.

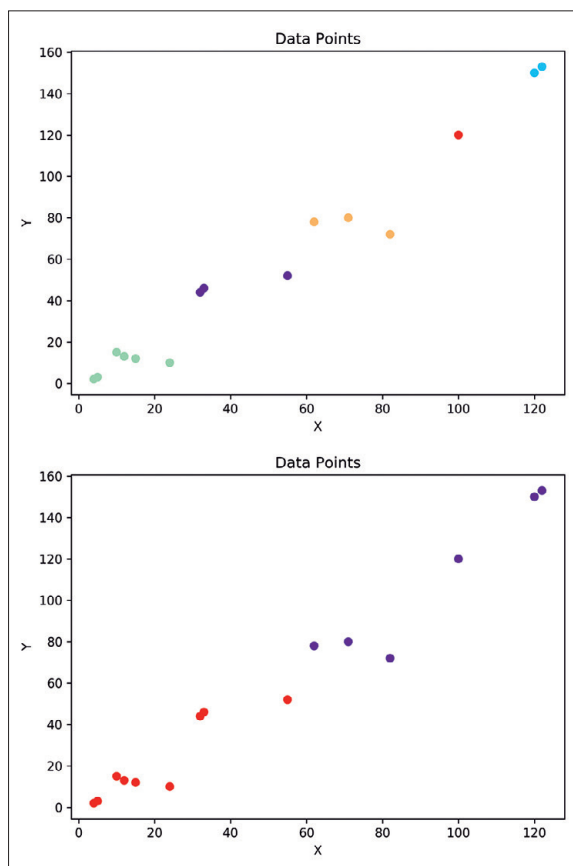
В файле `clustering.py` наиболее важен и интересен следующий код:

```
kmeans = KMeans(n_clusters=CLUSTERS)
kmeans.fit(DATA)
```

Две предыдущие команды применяют к данным алгоритм k-средних для создания необходимого количества кластеров, которое определено в переменной `CLUSTERS`. Получив данные, к ним можно применить алгоритм кластеризации k-средних с помощью всего двух команд Python 3 — это впечатляет! Файл `clustering.py` считывает данные из внешнего текстового файла, что делает скрипт еще более разносторонним. Надо только помнить, что в каждой строке текстового файла следует размещать два значения, разделенные запятой.

При запуске файла `clustering.py` будет получен следующий текстовый вывод, а также создастся изображение:

```
$. /clustering.py data.txt 2
Labels: [1 1 1 1 0 1 1 0 0 0 1 1 0 0]
$. /clustering.py data.txt 3
Labels: [2 2 2 2 1 0 0 0 0 0 2 1 1]
```



Два рисунка, созданные с помощью скрипта `clustering.py`. Этот скрипт использует модуль `scikit-learn` для кластеризации данных с использованием метода k-средних; данные считываются из внешнего текстового файла.

```
$. /clustering.py data.txt 10
Labels: [6 6 8 8 0 1 5 5 2 7 9 4 8 3 3]
```

Цифры в выходных данных означают номер кластера, к которому принадлежит точка данных, находящаяся в этом положении. На графиках внизу показаны два файла изображений, созданных файлом `clustering.py` при обработке текстового файла с 15 точками. Обратите внимание: если объем данных у вас невелик, то такое множество кластеров вам ни к чему.

Поприветствуйте TensorFlow

«О, я слышал об этом! — говорите вы. — *TensorFlow* — это открытая библиотека для искусственного интеллекта». Да, это так. Первое, что надо сделать — установить пакет `tf` для Python 3. Для этого проще всего обратиться к менеджеру пакетов `pip3`, выполнив команду `sudo pip3 install tensorflow`. Учтите, что `pip3` скорее всего установит гораздо больше пакетов, чем требуется пакету `tensorflow` для Python 3.

Чтобы определить используемую версию *TensorFlow* (она часто обновляется), выполните команду

```
$ python3 -c 'import tensorflow as tf; print(tf.__version__)'
1.5.0
```

Учтите, что в *scikit-learn* вы сначала объявляете объект нужного алгоритма, затем обучаете свою модель и можете получать прогнозы для тестового набора данных. Однако с *TensorFlow* всё немного иначе. Сначала вы определяете вычислительный граф, который создается путем объединения некоторых математических операций, поддерживаемых в *TensorFlow*, и затем инициализируете несколько переменных. Потом вы добавляете данные на графики с помощью переменных и с использованием местозаполнителя. После этого вы создаете сеанс, а затем передаете созданный ранее график сеансу, при этом запускается выполнение сеанса. Наконец, вы закрываете сеанс.

Без напряжения

Итак, пора создать простой пример использования пакета *tensorflow* на Python 3. Ниже представлена небольшая часть скрипта `tFlow.py`, в котором *TensorFlow* используется для создания нейронной сети:

```
a = tf.placeholder(tf.int16)
b = tf.placeholder(tf.int16)
addition = tf.add(a, b)
mul = tf.multiply(a, b)
init = tf.global_variables_initializer()
```

В этом коде объявляются два местозаполнителя `a` и `b` для двух переменных типа `int16` и затем объявляются еще две переменные с именами `addition` и `mul` для сложения и умножения двух чисел соответственно.

При запуске файла `tFlow.py` получится результат, приведенный на экранном снимке стр. 77 — этот скрипт просто складывает целые числа.

Расширяем наши возможности с Theano

Хотя графические изображения всегда впечатляют, иногда приходится выполнять математические вычисления. *Theano* — мощная библиотека для математических вычислений, которая может пригодиться, когда вам приходится иметь дело с многомерными массивами, содержащими большие или огромные объемы данных. В этом разделе кратко описано использование *Theano*. Естественно, первым делом нужно установить *Theano*, командой `sudo pip3 install Theano`.

Код Python 3 для этого раздела сохранен в файле `theanoUse.py` и выглядит следующим образом:

```
#!/usr/bin/env python3
import numpy
```



```

3. mtsouk@mail: ~ (bash)
iMac:code mtsouk$ ./tFlow.py
1 + 2 = 3      1 * 2 = 2
1 + 3 = 4      1 * 3 = 3
1 + 4 = 5      1 * 4 = 4
2 + 2 = 4      2 * 2 = 4
2 + 3 = 5      2 * 3 = 6
2 + 4 = 6      2 * 4 = 8
3 + 2 = 5      3 * 2 = 6
3 + 3 = 6      3 * 3 = 9
3 + 4 = 7      3 * 4 = 12
4 + 2 = 6      4 * 2 = 8
4 + 3 = 7      4 * 3 = 12
4 + 4 = 8      4 * 4 = 16
5 + 2 = 7      5 * 2 = 10
5 + 3 = 8      5 * 3 = 15
5 + 4 = 9      5 * 4 = 20
6 + 2 = 8      6 * 2 = 12
6 + 3 = 9      6 * 3 = 18
6 + 4 = 10     6 * 4 = 24
7 + 2 = 9      7 * 2 = 14
7 + 3 = 10     7 * 3 = 21
7 + 4 = 11     7 * 4 = 28

```

На этом экранном снимке изображен вывод скрипта **tFlow.py**, который использует библиотеку *TensorFlow* для сложения и умножения целых чисел. Несмотря на свою простоту, **tFlow.py** иллюстрирует весь цикл программы *TensorFlow*.

```

import theano.tensor as T
from theano import function
x = T.dmatrix('x')
y = T.dmatrix('y')
z = x + y
f = function([x, y], z)
i = f([[[-1, 1], [-2, 2]], [[8, -8], [12, -12]]])
print(i)
print(type(i))

```

После создания двух матриц **x** и **y** надо создать функцию **f** для их сложения. Обратите внимание, что в *Theano* необходимо вводить все символы. При запуске файла **theanoUse.py** вы получите следующий результат:

```

$ ./theanoUse.py
[[ 7. -7.]
 [ 10. -10.]]
<class 'numpy.ndarray'>

```

Последняя строка подтверждает, что переменная **i** является массивом **NumPy**.

Использование Keras

Чтобы погрузиться в библиотеку *Keras*, сначала установите ее с помощью команды `sudo pip3 install keras` и аналогичного метода. Использование библиотеки *Keras* иллюстрируется в файле **kerasUse.ph**. После запуска файла **kerasUse.py** вы получите следующий результат:

```

$ ./kerasUse.py
Using TensorFlow backend.
X_train: (60000, 28, 28) y_train: (60000,)
$ ls *.png
1.png 11.png 13.png 15.png 17.png 19.png
20.png 4.png 6.png 8.png
10.png 12.png 14.png 16.png 18.png 2.png
3.png 5.png 7.png 9.png

```

Как видно из результата запуска файла **kerasUse.py**, скрипт использует *TensorFlow* и базу данных MNIST. Дополнительную информацию о базе данных MNIST можно получить на странице https://en.wikipedia.org/wiki/MNIST_database.

Все созданные изображения были вручную объединены в одно изображение, которое можно увидеть на рисунке внизу справа. Обратите внимание, что в машинном зрении важно сначала проверить данные, построив их график, чтобы избежать глупых ошибок в будущем.

» ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ДАННЫХ

Предварительная обработка данных является важной частью машинного обучения, так как при наличии неправильных данных алгоритмы машинного обучения не смогут выполнить свою работу. Поэтому важно, чтобы данные были в правильном формате.

Два серьезных вопроса, которые часто возникают при работе с данными — что делать с отсутствующими данными и что делать с неправильными данными. Если в какой-либо записи отсутствуют данные, вы можете либо проигнорировать эту запись, либо заполнить отсутствующие данные значениями по умолчанию. Однако работа с неправильными данными намного сложнее, потому что вам надо решить, стоит ли игнорировать запись с неправильными данными или попытаться исправить данные, изучив запись более подробно. Вообще говоря, если данных у вас много, то проще и быстрее всего игнорировать записи с отсутствующими и неправильными значениями.

В предварительной обработке данных помогут различные инструменты, включая такие языки программирования, как AWK, Python, Perl, Go и Ruby — если вы четко знаете, чего хотите достичь. Например, у функции `numpy.genfromtxt()` есть параметр `filling_values`, который позволяет определить набор значений по умолчанию, используемых при отсутствии некоторых данных.

Таким образом, ключом к успешной обработке и предварительной обработке данных является четкое документирование процесса.

Что дальше...

Эта статья — лишь введение в машинное обучение с использованием Python 3. Если вы хотите подробнее узнать о машинном обучении и статистике, рекомендуем зайти на сайты Springer (www.springer.com) и Packt Publishing (www.packtpub.com). На сайтах этих издательств вы найдете много книг по данной теме.

Кроме того, вы можете ознакомиться с дополнительными сведениями о *TensorFlow*, открытом проекте Google, на сайтах www.tensorflow.org и <https://github.com/tensorflow>, а со сведениями о модуле *scikit-learn* — на сайте <http://scikit-learn.org>. В представленном коде также показано, что стандартные библиотеки Python 3, такие как *Matplotlib*, *NumPy* и *SciPy*, интенсивно используются библиотеками машинного обучения Python 3, поэтому их тоже стоит изучить.

Однако действительно важно то, насколько вы экспериментируете и сколько разных подходов применяете к своим данным, чтобы получить нужную информацию или удивительные результаты! Машинное обучение сможет изменить вашу жизнь и карьеру, поэтому спешите начать им пользоваться! **LXF**



Вывод скрипта **kerasUse.py** при запуске без аргументов. В этом примере используется набор данных MNIST, а также глубокое обучение и машинное зрение для распознавания символов.

WORDPRESS

Как создать свою тему для сайта

Кент Ельчук показывает, как взять почти любую тему сайта и преобразовать ее в шаблон Wordpress с массой преимуществ...



НАШ
ЭКСПЕРТ

Кент Ельчук разрабатывает и поддерживает приложения для различных учреждений, а также web-приложения.

На этом уроке мы рассмотрим, как создать собственную тему Wordpress на основе любого базового шаблона сайта или CSS-фреймворка для адаптивного web-дизайна, например, Bootstrap. Хотя мы подробно разъясним все этапы создания темы, этот процесс будет намного понятнее тем, у кого есть базовые знания HTML и адаптивного дизайна. Эта статья пригодится всем, кто хочет преобразовать существующий статический сайт (или любой другой сайт) в Wordpress.

Поскольку исходный код Wordpress открыт, сайт будет целиком принадлежать вам, и вы сможете бесплатно разместить его в своей домашней сети. И так как он доминирует на рынке CMS (Content Management System — система управления контентом), навыки по созданию пользовательских шаблонов пригодятся web-разработчикам и владельцам сайтов.

Если вам нужна еще одна причина, по которой стоит научиться настраивать Wordpress, подумайте вот о чем. Хотя владельцы сервисов по самостоятельному созданию сайтов, таких как Godaddy, Wix и Weebly, могут тратить много денег на рекламу, люди предпочитают открытый Wordpress. Вот некоторые статистические данные, подтверждающие это: https://w3techs.com/technologies/overview/content_management/all.

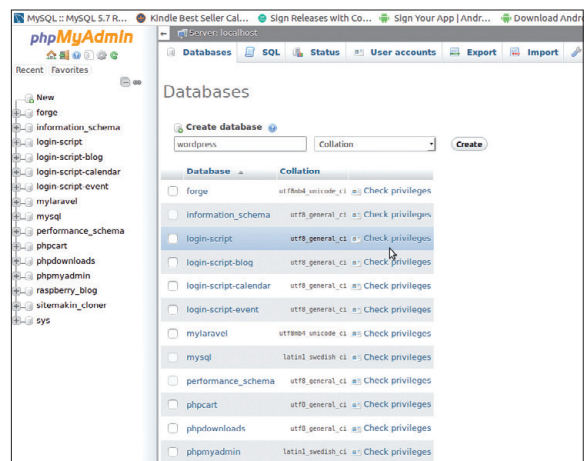
В отличие от конкурентов, в случае с Wordpress сайт и его код принадлежат вам. С другими сервисами по созданию сайтов вы чаще всего столкнетесь с проблемами при программировании. А решив отказаться от этих сервисов, вы сэкономите массу времени.

Мы постараемся сохранить наш учебник максимально простым, чтобы избежать всяких сложностей, и достаточно подробным, чтобы он остался информативным. В конце этого урока вы сможете создать собственное меню со ссылками на отдельные страницы или категории блога. Поскольку Wordpress обрабатывает различные типы содержимого (страницы и категории) по-разному, вам понадобится код для вывода выбранного содержимого.

Станьте администратором Wordpress

После входа в систему вы увидите множество функций и кнопок. Если это ваш первый опыт работы с Wordpress, не пугайтесь. Из нашего урока вы узнаете, как создать шаблон. Этот процесс включает настройку страниц и записей [post] для проверки правильной работы шаблона. Если всё хорошо, установка должна завершиться быстро, и на создание собственного файла **wp-config.php** потребуются не более пары минут.

После входа в панель администрирования Wordpress вы можете выбрать Appearance [Внешний вид] в левом столбце и затем



Создание базы данных MySQL в phpMyAdmin для установки Wordpress. Без нее невозможно создать сайт на Wordpress.

Themes [Темы]. После этого вы увидите несколько тем, таких как Twenty Eighteen, Twenty Seventeen, Twenty Sixteen и Twenty Fifteen.

Поскольку мы хотим создать собственный шаблон с нуля, мы добавим его к этому набору тем.

Настройка новой темы

Первым шагом при разработке новой темы является создание двух файлов — **style.css** и **index.php**. Основной файл для загрузки темы и требуемых файлов — **index.php**, а файл **style.css** можно использовать для создания стилей.

Эти два файла можно добавить в новый каталог, расположенный в каталоге **wp-content/themes**. Если заглянуть в этот каталог сейчас, вы увидите каталоги с другими темами, например, twentyseventeen. Для простоты назовем этот каталог **MyTheme** и отметим, что он содержит два файла: **index.php** и **style.css**.

По всей вероятности, у вас уже есть шаблон, который вы хотите использовать, и у этого шаблона уже есть структура файлов, которая формирует внешний вид шаблона. Чтобы перенести этот шаблон в Wordpress, можно открыть файл в браузере и скопировать исходный код в наш файл **index.php**.

После этого скопируйте все каталоги, содержащие CSS, код JavaScript, шрифты и другие ресурсы. Они могут находиться в папке под названием **assets**. На нашем уроке мы предполагаем, что в папке **assets** есть всё необходимое для отображения страниц.

В файл **style.css** надо добавить следующий код (имейте в виду, что этот файл обязателен для всех тем Wordpress). Файл CSS должен содержать несколько строк, как показано ниже. Не забудьте изменить имя автора и название темы.

```
/*
Theme Name: MyTheme
Author: Kent Elchuk
Description: Site template converted to WordPress
Version: 0.0.1
Tags: bootstrap
*/
```

Теперь перейдите в раздел Appearance > Themes [Внешний вид > Темы] и активируйте новую тему. Это позволяет одновременно создавать и тестировать темы, так как выбранная тема будет использоваться в качестве текущей темы сайта.

Пора проверить правильность ссылок на CSS и JavaScript.

Поскольку эти файлы будут находиться внутри каталога нашей темы, нужно добавить совсем немного кода перед старыми путями. Добавим следующий код в начало всех путей: `<?php echo get_bloginfo('template_directory');?>/`.

Это позволяет сравнить исходные и новые ссылки, как показано ниже.

Исходные ссылки:

```
<link href="assets/plugins/bootstrap/css/bootstrap.min.css"
rel="stylesheet" type="text/css"/>
```

Новые ссылки:

```
<link href="<?php echo get_bloginfo('template_directory');?>
/assets/plugins/bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css"/>
```

Если вы создали весь код HTML и расположили файлы в правильных каталогах, вы увидите ту же страницу, что и в простом HTML-файле. На самом деле, мы можем также открыть свой сайт Wordpress в браузере и проверить результаты.

Это была легкая часть работы. Теперь надо разделить **index.php** на три файла. Для этого мы вырежем код заголовка из файла **index.php** и вставим его в файл **header.php**, который находится в том же каталоге.

Вырезав код заголовка и вставив его в файл **header.php**, мы добавим код `<?php wp_head();?>` после закрывающего тега `</head>`. После этого добавим `<?php get_header();?>` в верхнюю часть файла **index.php**, откуда вы вырезали код заголовка.

Фрагменты кода, которые мы только что добавили, не требуют особых пояснений. Функция `wp_head()` сообщает Wordpress, что это заголовок. Назначение другой функции, `get_header()`, понятно по названию. По умолчанию Wordpress получает заголовок из файла с именем **header.php**.

Если заглянуть в другие шаблоны Wordpress, особенно в те, которые по умолчанию входят в Wordpress, вы увидите, что файлы **header.php** есть в папках большинства тем.

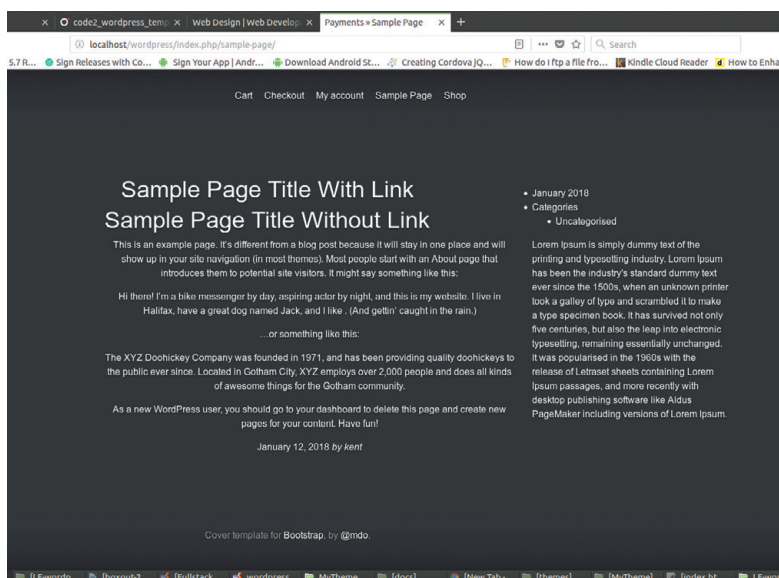
Заголовки страниц

По очевидным причинам, название страницы должно формироваться динамически каждый раз при открытии новой страницы в браузере. Код между тегами `title` можно добавить в заголовок вместо старых тегов `title`.

```
<title><?php bloginfo('name');?><?php wp_title();?>
</title>
```

После этого повторим процедуру с кодом нижнего колонтитула [footer]. При вставке кода в файл **footer.php** мы добавляем `<?php wp_footer();?>` над закрывающим тегом `</body>`. Затем мы добавляем `<?php get_footer();?>` в нижнюю часть файла **index.php**, из которой был вырезан код.

Эти две функции работают точно так же, как заголовок. В одной из них определяется код нижнего колонтитула, в другой — вызывается.



Для создания пользовательской темы нужно создать новую папку с несколькими файлами в каталоге **wp-content/themes**. Для начала потребуются всего два файла: **index.php** и **style.css**.

Сделав это, можно обновить страницу и проверить, насколько она соответствует исходному HTML-файлу. Соответствие должно быть точным. Если это не так, необходимо проверить синтаксис и ссылки. Помните, что для всех ссылок необходимо указать префикс `<?php echo get_bloginfo('template_directory');?>/`.

Учтите, что сделать это надо не только для ссылок в CSS и JavaScript, но и для ссылок на изображения. Таким образом, ссылка `` должна стать такой: ``.

Если надо добавить боковую панель, можете вырезать ее из файла **index.php** и переместить в новый файл **sidebar.php**. Для получения данных можно использовать функцию `<?php get_sidebar();?>` в том же месте, чтобы вырезать код из файла **index.php**. Использование фреймворка для адаптивного

» СТАВИМ WORDPRESS

После загрузки и распаковки Wordpress мы можем переместить эту папку на любой сервер, на котором мы хотим разместить сайт Wordpress. Можно использовать учетную запись web-хостинга или создать сайт на собственном локальном web-сервере, расположенном на компьютере Linux.

Настроив базу данных, откройте папку, для которой выполняется установка, например, `localhost/wordpress` или `http://example.com`. Нажмите кнопку **Run the Installation** [Запустить установку] и введите имя базы данных, имя пользователя и пароль. Префикс по умолчанию — **wp_**, но при желании можно указать другой. На некоторых хостингах всё это работает отлично. Если на домашнем сервере не удастся записать конфигурационный файл, нужно изменить файл **wp-config-sample.php**, добавив в него имя базы данных, имя пользователя и пароль, и затем сохранить этот файл под именем **wp-config.php**.

В конце установки добавьте название сайта, имя пользователя и пароль для панели администрирования. Не путайте эти имя пользователя и пароль с именем пользователя и паролем для базы данных. С помощью этих имени пользователя и пароля мы можем войти на сайт и управлять содержимым, темой и другими функциями сайта в браузере.

После ввода этих данных и нажатия кнопки **Install Wordpress** [Установить Wordpress] открывается окно входа в панель администратора. Введите эти имя пользователя и пароль и войдите в систему.



дизайна, такого как Bootstrap, позволяет легко использовать один столбец в качестве содержимого и более узкий столбец для боковой колонки.

Теперь перейдем к содержимому. Учитывая ширину, содержимое — это всё, что нам захочется. Например, если мы используем боковую панель Bootstrap с классом col-md-4, его содержимое будет содержимым внутри соответствующего столбца в строке с именем colmd-8.

Ниже приведен пример типичной строки с двумя столбцами. Меньший столбец находится справа. Мы используем классы 'row' и 'colmd-X' из Bootstrap, так как это самый популярный фреймворк для адаптивного web-дизайна, и созданная тема скорее всего также будет адаптивной.

Как видите, для того, чтобы добавить основное содержимое и боковую панель, требуется всего четыре простых строки кода:

```
<div class="row">
<div class="col-md-8"></div>
<div class="col-md-4"></div>
</div>
```

На нашем уроке два указанных столбца упростят нам обсуждение, разделив содержимое столбца и боковой панели.

Широкий столбец `<div class="col-md-8"></div>` будет находиться в файле `index.php`. Этот файл можно использовать только для вывода данных. С другой стороны, мы можем включить в него другие части.

Поэтому, чтобы загрузить дополнительное содержимое в `index.php`, мы просто добавим `<?php get_template_part('content', get_post_format());?>`. Таким образом, простой раздел с содержимым в файле `index.php` будет выглядеть так, как далее показано в коде. Включаемый файл будет называться `content.php` и располагаться в той же папке:

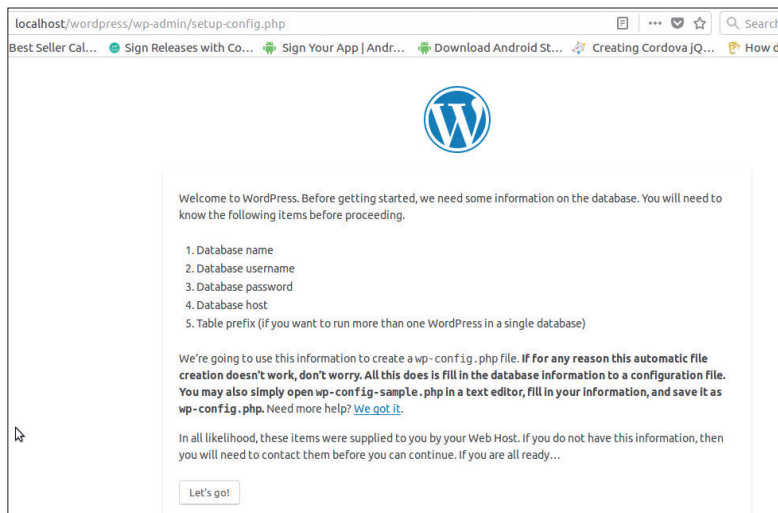
```
<div class="col-md-8">
<?php get_template_part( 'content', get_post_format() );?>
</div>
```

Создавать файл `content.php` и использовать функцию `get_template_part('content', get_post_format());` не обязательно. Однако такой подход позволяет эффективно разбить файлы на фрагменты меньшего размера.

Между тем, после создания файла `content.php` мы можем добавить в оба файла код для загрузки динамических сообщений и страниц. Ниже показана последняя порция кода, добавленная в файл `index.php`.

```
<div class="col-md-8">
<?php
if ( have_posts() ) : while ( have_posts() ) : the_post();
```

Распаковав Wordpress, вы можете открыть сайт в браузере и следовать инструкциям по установке.



```
get_template_part( 'content', get_post_format()
);
endwhile; endif;
?>
</div>
```

Добавляем содержимое

Содержимое надо добавить в файл `content.php`. Например, приведенный ниже код выводит заголовок в верхней части страницы в тегах `<h1></h1>` со ссылкой обратно на исходную запись. Тогда, щелкнув по категории и увидев список, вы сможете перейти к конкретной записи:

```
<div class="row" style="background:none;">
<div>
<div class="col-md-12">
<a href="<?php the_permalink();?>"><h1 class="blog-post-title"><?php the_title();? </h1></a>
<h1 class="blog-post-title"><?php the_title();?></h1>
</div>
</div>
</div>
<!-- POST BEGIN -->
<div class="blog-post">
<?php the_content();?>
<p class="blog-post-meta"><?php the_date();?> <em>by
<a href="#"><?php the_author();? </em></a></p>
</div>
<!-- POST END -->
```

Добавляем боковую панель

Сейчас мы можем создать статическую боковую панель для вашего сайта. В боковой панели можно загружать обычные ссылки Wordpress (например, категории и архивы по месяцам) с помощью встроенных функций Wordpress.

Поскольку содержимое было вставлено в столбец с классом Bootstrap col-md-8, мы можем добавить код в файл `sidebar.php` в столбец с классом col-md-4. В боковой панели будут указаны ссылки на архивы по месяцам и на категории блога. В следующих двух строках кода показано, как это сделать.

```
<!-- BEGIN SIDEBAR -->
<div class="col-md-4">
<?php wp_get_archives( 'type=monthly' );?>
<?php wp_list_categories();?>
</div>
```

Сейчас мы еще раз опишем настройку содержимого и боковой панели в несколько других терминах на случай путаницы. Класс Bootstrap row (class="row") представляет собой элемент `div`, включающий основное содержимое и боковую колонку, которые представляют собой два блока `div`.

Основное содержимое будет находиться в файле `content.php`, а содержимое боковой панели — в файле `sidebar.php`. Идея состоит в том, что функция `<?php get_sidebar();?>` располагается сразу перед закрывающим тегом `</div>` для открывающего тега `<div class="row">`.

Изменив файл `index.php`, можно обновить сайт, и мы увидим содержимое записи. Если сайт совершенно новый, мы увидим содержимое записи "Hello World", потому что в Wordpress по умолчанию отображаются все записи.

Добавим еще одну запись и откроем ее URL-адрес в браузере. Таким образом, если мы создадим новую запись с именем Test, то сможем просмотреть страницу и увидеть новый заголовок и содержимое, которые мы создали в редакторе. Если вы забыли, как добавить запись, напомним: выберите Posts > Add New [Записи > Добавить новую]. Теперь мы можем добавить заголовок и некоторое содержимое и сохранить запись. Страница создается точно

так же, надо лишь выбрать Pages > Add New [Страницы > Добавить новую].

Однако обычно пользователь хочет отобразить страницу. Чтобы установить страницу в качестве домашней, перейдите в раздел Themes > Customize > Homepage Settings [Темы > Настройка > Параметры домашней страницы]. В этом разделе вы увидите несколько параметров. Чтобы сделать страницу домашней, просто нажмите радиокнопку A static page [Статическая страница] и затем выберите нашу страницу.

А сейчас надо решить еще одну важную задачу: разобраться, как создать динамическое меню. Выберите Themes > Customize > Create New Menu [Темы > Настройка > Создать меню], назовите меню 'Main' и выберите Next > Add Items > Select From List > Publish [Далее > Добавить элементы > Выбрать из списка > Опубликовать].

На следующем этапе работы над меню мы создаем новый файл с именем **functions.php** и будем работать с ним. С помощью приведенного ниже кода вы можете добавить свое меню в меню Appearance [Внешний вид], которое включает пункты Themes [Темы], Customize [Настройка] и Editor [Редактор]. Сохранив файл **functions.php**, вы можете создать меню.

```
function register_my_menu() {
    register_nav_menu('main',__( 'Main' ));
}
add_action( 'init', 'register_my_menu' );
```

Создав меню, откройте Menus > Manage Locations [Меню > Управление расположениями] и назначьте меню Main расположению темы [Theme Location], которое также называется Main. После этого сохраните изменения. Чтобы использовать это меню, просто откройте файл **header.php** и добавьте в него следующий код:

```
<?php wp_nav_menu( array( 'theme_location' => 'main' ) ); ?>
```

После добавления кода в файл **header.php** вы увидите это меню, но оно вряд ли будет выглядеть так, как нам хотелось. К счастью, это легко исправить.

Часто главное меню представляет собой неупорядоченный список. Чтобы это исправить, можно заменить предыдущую строку кода на следующую строку. В этой строке отсутствует ненужный **div** вокруг неупорядоченного списка, а список оборачивается в тег **ul** с классом. Если теперь обновить страницу, вы увидите, что меню выглядит так же, как исходное, и содержит нужные пункты:

```
<?php wp_nav_menu( array( 'theme_location' => 'main',
'container' => false, 'items_wrap' => '<ul class="nav">%3$s</ul>'
) ); ?>
```

И еще один шаг. Нам может понадобиться активный класс для страницы, которая загружена в браузере. Просто добавьте следующий код в файл **functions.php**:

```
add_filter( 'nav_menu_css_class', 'special_nav_class', 10, 2);
function special_nav_class( $classes, $item ) {
    if ( in_array( 'current-menu-item', $classes ) ) {
        $classes[] = 'active';
    }
}
return $classes;
}
```

Дополнения к навигации

Ну вот, мы обзавелись довольно удобным работающим сайтом на Wordpress. Мы можем добавлять записи в блог на лету, управлять меню, публиковать новые страницы и редактировать любое содержимое. Теперь было бы полезно описать структуру папок нового сайта Wordpress и шаблона.

» ОТОБРАЖЕНИЕ ЗАПИСЕЙ ПО КАТЕГОРИЯМ

Для управления категориями выберите Posts > Categories [Записи > Категории] из меню слева в разделе администрирования. Мы можем добавить пункт меню, который отображает записи для заданной категории. В данном случае наша категория называется web-development-and-seo. Описательная часть ссылки [slug] берется из категории.

Если web-страница представляет собой запись, возвращается ее содержимое. Если это не так, возвращаются все записи из категории и отображается список заголовков со ссылками на эти записи. Код для вывода списка записей в категории покажется вам знакомым, за исключением того, что этот код следует изменить в файле **index.php**, так как мы хотим создать правило, отделяющее содержимое записи от выбранной категории.

```
<?php
if (have_posts()) : while (have_posts()) : the_post();
    get_template_part('content', get_post_format());
endwhile;
else:
    $cat_id = get_query_var('web-development-and-seo');
    $paged = ( get_query_var( 'paged' ) ) ? get_query_var( 'paged' ) : 1;
    $catquery = new WP_Query('cat=web-development-andseo&posts_per_page=5&paged=' . $paged);
    ?>
<ul>
<?php while ($catquery->have_posts()) : $catquery->the_post(); ?>
    <li><a href="<?php the_permalink() ?>" rel="bookmark"><?php the_title();
?></a></li>
<?php endwhile; ?>
<?php
next_posts_link( 'Older Entries', $catquery->max_num_pages );
previous_posts_link( 'Newer Entries' );
?>
<?php wp_reset_postdata();
endif;
?>
</div>
<?php get_sidebar(); ?>
```

В этой статье мы просто установили Wordpress, настроили его, и всё работает. А если у нас уже есть рабочий сайт, и мы создали тему для определенного подкаталога?

Это немного усложняет ситуацию. Во-первых, так как мы хотим использовать основной сайт и новый подкаталог Wordpress, мы можем жестко скопировать ссылки с основного сайта в файл **header.php**. Так мы сохраним исходную навигацию.

Поскольку основной сайт находится уровнем ниже, мы можем добавить тег **base** в заголовок, чтобы ссылки из Wordpress указывали на корневой каталог сайта. Эта строка показана ниже:

```
<base href="/" />
```

Мы можем проявить креативность и по-прежнему использовать сочетание жестких ссылок и динамических пунктов в меню, хотя лучше сохранить структуру однородной без жестко закодированных ссылок.

Заключение

На этом всё. Теперь вы можете интегрировать блог в большинство сайтов или быстро создать пользовательскую тему. Удачной работы над шаблонами! **LXF**

» ПОДПИШИТЕСЬ НА ЖУРНАЛ LINUX FORMAT на www.linuxformat.ru/subscribe/

Новое поколение средств защиты

Межсетевые экраны ССПТ, не имеющие IP-адреса

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

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

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

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

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

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

Дополнительное устройство защиты для:

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

Москва
+7 (499)

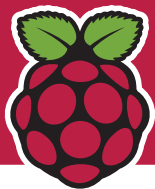
271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center



Лес Паундер рад порассуждать на любую тему о Raspberry Pi под кружку пива с пакетиком соевых орешков.

» СЧАСТЛИВОГО ДНЯ ВАРЕНЬЯ!

В момент написания статьи я нахожусь на старте уикэнда празднования шестилетия Raspberry Pi. На Raspberry Jam этого уикэнда в Престоне, вместе со многими другими во всём мире, будет отмечаться успех, родившийся в 2012 г. Пускай Raspberry Pi из Кембриджа, а сообщество Jam основано где-то на Северо-Западе Англии — одни говорят, что в Престоне, другие — что в Манчестере; но оба выросли от скромных корней.

В самом начале было очень мало штук Raspberry Pi и не слишком много информации в помощь пользователям. Джеммы (встречи) дают энтузиастам время и доступ к устройствам, достаточные для разработки проектов, блогов и документов, которые достигли положения стандартов де-факто.

Я присутствовал на обоих первых Джеммах. Алан О'Донохоу [Alan O'Donohoe] возглавлял Джем в Престоне, он использовал свой опыт работы с Raspberry Pi для помощи учителям в понимании кода. Джем в Манчестере возглавил Бен Наттолл [Ben Nuttall], ставший менеджером сообщества Raspberry Pi. Бен помог Эми Мезер [Amy Mather] научиться кодированию, а в 2013 г. благодаря своему упорству и наставничеству она выиграла конкурс Digital Girl of the Year [Цифровая девушка года].

Также приятно видеть молодых людей, выросших на Raspberry Pi, делающих свои первые шаги в кодировании, вдохновляющих других и разрабатывающих проекты, которые продолжают вдохновлять и помогать другим по всему миру. Это сообщество тех, кто не только пользуется устройством и аксессуарами, а еще и создает новые и важные проекты. Именно этому и служат Raspberry Pi и открытый исходный код.

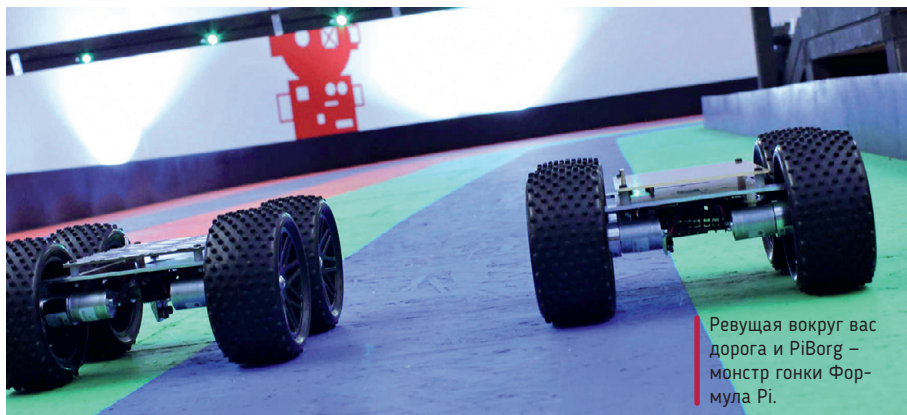
Формула Pi выходит на скоростную полосу

Pi-двигатели, от винта: сезон 2018 г. открывается увлекательными классами и новыми правилами гонки.

Гонки Формулы Pi 2018 должны состояться в июне, а регистрация закончилась 30 апреля.

Формула Pi вступила в жизнь как проект на платформе Kickstarter еще в июле 2016 г., причем его финансирование зашкалило за 300% от намеченной цели. За успешным зимним

сезоном 2016 г. последовали летний и зимний 2017 г., а теперь уже выстраивается сезон 2018 г. Оригинальный класс Grand Pi заменен классом Monster, и должен начать тестироваться новый высокоскоростной класс Raptor с ограниченным участием. Узнайте больше и участвуйте на www.formulapi.com.



Ревущая вокруг вас дорога и PiBorg — монстр гонки Формула Pi.

Pipecam

Сделай подводные снимки.

Дайвер Фред Фури [Fred Fourie] разработал корпус подводной трубы для камеры на базе Pi. Не теряя работоспособности вплоть до давления в 4 бара, она может снимать на глубине до 30 м в герметичной трубе, где также предусмотрен аккумулятор для питания. Идеально подходит для рядовых дайверов; вы можете узнать больше на <https://hackaday.io/FredWFourie>.



Шпирн за вашими осетровыми.

OpenCat

Не этот ли пушистик питомец Pi?

Довольно впечатляющая сборка создает робота, работающего на Pi3. Знакомьтесь с Peto, домашним котом, отпечатанным на 3D-принтере. Его создатель, Ронгчжонг Ли [Rongzhong Li], делал этот проект с целью обучающего опыта, и конечный результат впечатляет. Узнайте больше о входах и выходах робота из семейства кошачьих, посетив www.hackster.io/petoi/opencat-845129.



Не убивайте птичек, для этого есть робот.

Raspberry Pi 3 B+

Лес Паундер обнаруживает еще кусочек Pi. И этот инкрементно лучше предыдущего, но по нему ли мы изголодались?..

ВКРАТЦЕ

Самый последний в серии Raspberry Pi, на сей раз улучшивший работу в сети (802.11ac и "Gigabit" через USB 2.0). Скорость процессора немного повысилась, но ОЗУ остается таким же, как в предыдущей модели. Теперь Pi 3+ поставляется с PoE через отведенные контакты GPIO, но сохраняет совместимость со стандартным 40-контактным GPIO.

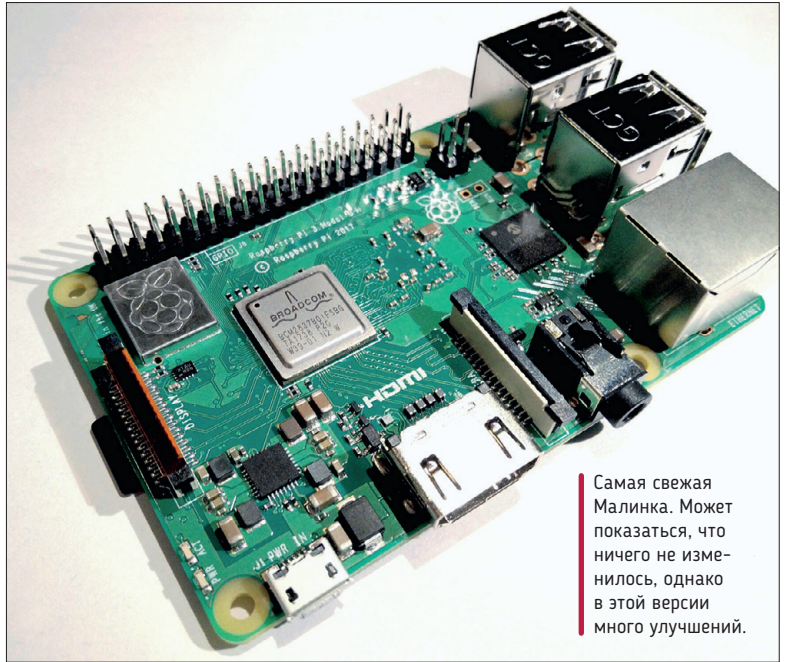
Еще один Raspberry Pi! Будет ли это тот самый пресловутый Raspberry Pi 4, предлагающий USB 3.0, SATA, GTX 1080 GPU, Octa Core CPU и 16 ГБ ОЗУ?

Сначала ответим на самые очевидные вопросы. Нет, это не Raspberry Pi 4, и нет, в нем нет никаких этих функций. Скорее Raspberry Pi 3+ является инкрементным улучшением Pi 3 B, предлагая чуть больше мощности и намного лучшие возможности подключения. Это не первая плата B+ — то была оригинальная B+, выпущенная в 2014 г.; при ней дизайн первой платы превратился в фактический стандарт раскладки плат, который часто повторяют, но никак не могут превзойти конкуренты.

Инкрементное обновление

Давайте начнем с System on a Chip (SOC): это BCM2837B0, содержащий CPU. Как и в Pi 3, мы имеем 64-битный CPU ARM Cortex A53, но теперь работающий на 1,4 ГГц, что на 200 МГц больше, чем у Pi 3. Разница не столь уж огромная, но помогает сделать работу Raspbian немного ровнее. В общем Raspbian стал откликаться лучше, и использовать его теперь — настоящее удовольствие, так что дополнительная мощность свое дело сделала.

ОЗУ по-прежнему 1 ГБ. Этого достаточно для большинства опытных пользователей Pi, однако было бы неплохо иметь 2 ГБ ОЗУ, что быстро становится нормой благодаря таким платам, как Asus Tinkerboard. Мы запустили тест *sysbench*, который находит



Самая свежая Малинка. Может показаться, что ничего не изменилось, однако в этой версии много улучшений.

все простые числа до 10000, используя все четыре ядра CPU. Pi 3 B+ сделал это за 36,583 секунд против 45,7046 для исходного Pi 3. Это почти на 20% быстрее!

Улучшенная работа в сети

С Pi 3 в 2016 г. мы увидели встраивание в плату Wi-Fi и Bluetooth, и это было далее улучшено для Pi 3+, где Wi-Fi теперь предлагает 5-ГГц 802.11ac, с намного большей пропускной способностью — теоретически до 1,3 Гб в секунду, но в тестах мы обнаружили, что она составляет 74 Мб/с (9,25 МБ/с) против стандартной скорости 802.11n — 47,8 Мб/с (5,875 МБ/с).

Беря конструктивную особенность из Pi Zero W, мы обнаружили, что беспроводная антенна имеет тот же дизайн, по лицензии от Proant AB. Подключения ко внешней антенне нет. Эта новая 5-ГГц опция Wi-Fi работает поразительно хорошо, и в наших тестах мы смогли посмотреть видео на YouTube с разрешением 1080p всего лишь с небольшой задержкой в начале воспроизведения.

Еще одно улучшение касается Ethernet. В предыдущих моделях Pi мы видели чипы контроллера LAN9512 и LAN9514, которые предлагали порты USB 2.0 и подключение Ethernet. Для Pi 3 B+ мы видим новый LAN7515, предлагающий Gigabit Ethernet в интерфейсе USB 2.0, и с этим новым чипом Ethernet работает на 325 Мб/с (40,6 МБ/с) — солидное улучшение производительности, более чем в три раза по сравнению с предыдущим Pi 3, имевшим полосу пропускания на 94,3 Мб/с (11,78 МБ/с).

По сравнению с Asus Tinkerboard, которая использует чип RTL8211E для настоящего подключения Gigabit (в тестах мы зафиксировали 939 Мб/с, что составляет 117,375 МБ/с), Pi 3 B+ все еще тормозит. Однако для общих операций и применения

» СЕМЕЙНОЕ СХОДСТВО

Когда Raspberry Pi впервые вышел на сцену в 2012 г., у него был уникальный форм-фактор с единственным недостатком: плохим расположением отверстий для винтов. Фактически, в первой версии отверстий для винтов не было.

Но почему отверстия для винтов так важны? Дополнительные платы для Raspberry Pi бывают самых разных форм и размеров, и это создавало проблемы пользователям. Например, плата Pibrella была разработана для 26-контактного GPIO, и у нее была резиновая ножка, чтобы при нажатии ее не коротил порт HDMI. Требовалось стандартное соединение, и в Raspberry Pi B+ в 2014 мы увидели 40-контактный GPIO и отверстия для винтов по углам платы. Это обеспечивало удобную рабочую точку, где можно было закрепить платы на опорах M3.

Этот новый стандарт был частью HAT (Hardware Attached on Top), который подарил нам чипы EEPROM, способные обращаться с новым 40-контактным GPIO и позволяющие настроить плату. Потом это стало фактическим форм-фактором Pi, и последующая модель B следовала этой структуре, позволяя платам, созданным по данному стандарту, работать с рядом совместимых Raspberry Pi.

в качестве домашнего файлового сервера это долгожданное увеличение производительности, с достаточной полосой пропускания для потоковой передачи файлов на приличной скорости. Bluetooth также улучшен до BLE 4.2.

Мощь в Ethernet

Ставший стандартным 40-контактный GPIO (General Purpose Input Output — ввод/вывод общего назначения) остался прежним, обеспечивая совместимость со множеством имеющихся плат расширения. Мы протестировали GPIO с помощью традиционных библиотек RPi.GPIO Python и GPIO Zero, и обнаружили, что все хорошо, в том числе и использование дополнительных плат.

Совсем рядом с контактами GPIO находятся четыре дополнительных контакта, помеченные как PoE (Power over Ethernet), которые позволяют Pi 3+ получать питание и общаться через один и тот же провод, через будущий PoE HAT. Но здесь возникает одна маленькая проблема. Размещение контактов PoE означает, что их будет закрывать любая плата спецификации HAT, помещенная на GPIO. Мы предположили, что новый PoE HAT будет иметь сквозное соединение GPIO, но на момент написания у нас не было никакой информации. Контакты PoE дают возможность для некоторых плат напрямую контактировать с выводами, что приводит к риску короткого замыкания компонентов на нижней стороне плат. Советуем перед включением проверить зазор.

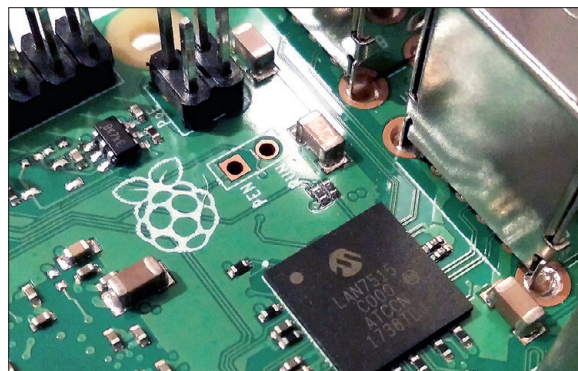
Унаследованные порты

У Pi 3 B+ такие же порты, как у Pi 3 (HDMI, официальный порт камеры/дисплея и комбинированный разъем аудио/видео). По части USB изменений нет, четыре порта USB 2.0 предлагают достаточно возможностей подключения для большинства пользователей. Питание подается через порт microUSB, и рекомендуется официальный источник на 5 В и 2,5 А.

В режиме простоя Pi 3 B+ может потреблять около 530 мА, однако при просмотре видео с YouTube в полноэкранном режиме этот показатель вырос до 1 А, и мы увидели устрашающий значок с молнией на рабочем столе, говоривший о том, что наш источник питания неподходящий. Поэтому мы перешли на официальный источник питания, который, к счастью, решил нашу проблему.

Лучшая терморегуляция

Нельзя отрицать, что Pi 3 сильно грелся. Мы подвергали наше устройство стрессовой нагрузке, воспроизводя видео с YouTube, летая в Minecraft и рассчитывая простые числа в *sysbench*, и оно нагрелось до 81,1°C! Но в Pi 3 B+ наивысшая температура, которой нам удалось достичь, была 67,1°C. Благодаря новому термальному колпачку, который помогает рассеивать тепло, Pi 3 B+ идеален для использования в качестве домашнего медиа-плеера, который обычно помещается в корпус.



При Wi-Fi в виде 802.11ac, 802.11b/g/n и Gigabit Ethernet через USB 2.0, Raspberry Pi 3+ — идеальная плата для сетевых проектов, например, файловых и медиа-серверов.

Сравнительный тест	Raspberry Pi 3	Raspberry Pi 3 B+
Ethernet (Мб/с)	94,3	325
2,4 ГГц (Мб/с)	35,6	47,8
5 ГГц (Мб/с)	—	74,7
Sysbench 4 ядра	36,5	31,8
Octane	2465	2780

Замена настольному ПК

Некоторые пользователи рассматривают Raspberry Pi не как платформу для создания чего-то или программирования, а как дешевую альтернативу обычному компьютеру. В предыдущих версиях это было возможно при соблюдении некоторых компромиссов, в основном касательно просмотра сайтов в Сети. Но поскольку браузером по умолчанию выбран *Chromium*, мы наблюдали повышение производительности обработки данных и теперь, с улучшенным проводным/беспроводным соединением, особенно в Pi 3, наш Raspberry Pi легко использовать взамен старого компьютера. Не забудем, конечно же, о дополнительном бонусе — он работает очень тихо и потребляет куда меньше энергии, чем ноутбук или настольный ПК.

Может, Raspberry Pi 3 B+ и не “Raspberry Pi 4”, но он никогда к этому и не стремился. Зато это инкрементное улучшение предлагает нам усовершенствования существующей комплектации. Повышенная мощность CPU радует, но самое большое улучшение — это работа в сети. Предложение чего-то большего, чем 100 Мб/с, как в старых моделях, через Gigabit через USB2 пригодится пользователям, жаждущим создавать сетевые устройства, а для тех, у кого имеются роутеры 802.11ac, 5-ГГц Wi-Fi является превосходным компромиссом скорости и портативности.

Raspberry Pi 3 B+ заслуживает большой любви. Он предлагает новые функции и улучшения, оставаясь верным своим корням. GPIO тот же, что и в предыдущих моделях (весьма иронично, начиная с B+ и выше), но примите во внимание контакты PoE, и убедитесь, что ваша плата их не касается.

Цена Raspberry Pi 3 B+ не превышает цену оригинального Pi 3, и если вы — новичок в Pi, эту модель определенно стоит купить. Однако владельцам существующих Pi эту покупку придется оправдывать.

И в заключение, это — отличная плата, достойный преемник Pi 3 и шаг по пути, который — мы надеемся — приведет нас к Raspberry Pi 4. **LXF**

Новый Pi предлагает солидное увеличение скорости CPU и большой скачок в передаче по проводным сетям.

ВЕРДИКТ

РАЗРАБОТЧИК: Raspberry Pi Foundation
САЙТ: www.raspberrypi.org
ЦЕНА: £ 35

ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
УДОБСТВО В РАБОТЕ	8/10
ОПРАВДАНОСТЬ ЦЕНЫ	10/10

Небольшое повышение мощности CPU при массе улучшений работы в сети означают, что этот Pi будет как дома и на вашем столе, и внутри робота. Отличное усовершенствование классики.

» **Рейтинг 9/10**

ФАЙЛОВЫЙ СЕРВЕР

Файловый сервер с блокировкой рекламы

Лес Паундер употребил мощь нового Raspberry Pi 3 B+ для создания файлового сервера и блокировщика рекламы в одном компактном устройстве!



НАШ ЭКСПЕРТ

Лес Паундер — умелец-фрилансер, который работает с такими организациями, как Raspberry Pi Foundation, чтобы продвигать практические навыки умельцев. Он ведет блог на www.bigl.es.

ВАМ НУЖНЫ

- > Raspberry Pi 3 или 3 B+
- > Интернет-соединение
- > Флешка или жесткий диск USB
- > Клавиатура, мышь и экран для вашего Pi
- > Официальный источник питания 2,5 А
- > Другой компьютер с Linux или Windows

Ч чтобы отметить релиз нового Raspberry Pi 3 B+, мы будем использовать его для создания нашего собственного сервера *Samba*, чтобы хранить файлы и медиа для доступа через нашу домашнюю сеть. Затем тот же Pi мы используем для блокировки рекламы и улучшения навигации.

Файловый сервер Samba

Давайте обеспечим себе последнюю версию Raspbian. Скачайте ее с сайта Raspberry Pi и с помощью вашего любимого инструмента запишите образ на пустую карту microSD. По завершении извлеките карту и поместите ее в свой Pi, затем подключите клавиатуру, мышь, экран, Ethernet и, наконец, питание. Ваш Pi загрузится, и поскольку это первая загрузка, расширит файловую систему, чтобы полностью использовать место на вашей SD-карте.

Вскоре вы увидите рабочий стол Raspbian, так что давайте обновим наш Pi. Откройте терминал; иконка находится в верхней левой части экрана. В терминале построчно введите написанное ниже и нажмите Enter в конце каждой строки:

```
$ sudo apt update
$ sudo apt upgrade -y
```

Чтобы мы всегда могли найти свой Pi в сети, исправим его IP-адрес. Щелкните правой кнопкой на иконке сети и выберите Wireless and Wired Network Settings [Настройки проводной и беспроводной сети]. В диалогe Network Preferences [Сетевые настройки] настройте интерфейс eth0. Отметьте Automatically configure empty options [Автоматическая настройка пустых опций] и смените адрес IPv4 в соответствии с вашей сетью. Например, наша сеть использует 192.168.0.xxx, так что мы выбрали IP-адрес как 192.168.0.100. Нажмите Apply [Принять], а затем Close [Закреть]. Затем перезагрузите Pi, чтобы убедиться, что IP-адрес изменился; для этого откройте терминал и введите

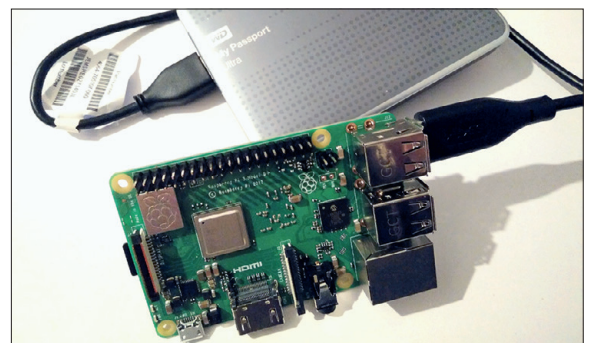
```
$ hostname -l
```

Давайте оставим терминал открытым и установим пакет *Samba*. Введите следующее и нажмите Enter:

```
$ sudo apt install samba samba-common-bin
```

Когда всё установится, вставьте USB-устройство для хранения, которое мы будем использовать в качестве общей памяти. Raspbian автоматически смонтирует диск для нас, но нам необходимо знать, где он будет смонтирован. В терминале введите команду

```
$ mount | grep /dev/sd
```



Наш простой сервер/блокировщик рекламы *Samba* питается через новый Pi3 B+, так что у нас есть экономный энергоэффективный прибор для всех домашних нужд.

Теперь (если вставлен только один USB-носитель) мы должны увидеть устройство с названием `/dev/sda1`. Это наш USB-носитель, так что запишите у себя устройство и где оно смонтировано — обычно в `/media/pi/`. Запомните полный путь к диску. Итак, у нас есть устройство — наш USB-диск, и на нем есть директория, к которой мы хотим сделать общий доступ через *Samba*. Давайте сообщим *Samba*, где найти диск. В терминале введите следующее, чтобы отредактировать файл `smb.conf`:

```
$ sudo nano /etc/samba/smb.conf
```

Когда файл загрузится, перейдите в нижнюю часть файла и воссоздайте следующие строки:

```
[share]
Comment = Pi shared folder
Path = /media/pi/<NAME OF DRIVE>
Browseable = yes
Writeable = Yes
only guest = no
create mask = 0777
directory mask = 0777
Public = yes
Guest ok = yes
```

Чтобы сохранить файл и выйти из редактора, нажмите Ctrl+O, затем Enter, а затем Ctrl+X. Теперь давайте убедимся, что пользователь по умолчанию, pi, добавлен в качестве пользователя *Samba*. В терминале введите


```
$ sudo smbpasswd -a pi
```

Далее, чтобы убедиться, что наша настройка Samba верна и готова к использованию, надо велеть *Samba* перезагрузиться, так что введите в терминале

```
$ sudo /etc/init.d/samba restart
```

По завершении установки перейдем к нашему компьютеру Windows или Linux и проверим, что мы можем найти и использовать сервер *Samba*.

В Ubuntu можно использовать файловый менеджер (*Nautilus*) для подключения через ярлык *Network*. Имя нашего сервера RASPBERRYPI должно быть видно. Нажмите “share” и, когда будет предложено, введите имя пользователя “pi” и пароль “raspberry” для доступа к серверу. Данный процесс выглядит похожим образом для устройств Windows 10, поскольку ярлык *Network* также можно найти в приложении *File Manager* [Файловый менеджер]. Пользователи Android также могут скачать приложение *Samba* из магазина приложений и получить доступ к файлам, прямо как десктоп-коллеги.

Итак, теперь у нас есть центральный пункт, где мы можем сохранять наши файлы и делиться ими через сеть — удобно для большого объема медиа, потребляемого людными домами!

Настройка Pi Hole

Все мы не любим рекламу на web-страницах, но установка блокировщика рекламы на каждое устройство весьма затратна по времени. Не создать ли одно устройство, которое отфильтрует всё это для нас? Pi-Hole — как раз такое устройство. Работая на нашем Raspberry Pi, Pi-Hole заменяет наш стандартный DNS-сервер (обычно — нашего ISP, если мы не поменяем это) на свой собственный, который отфильтровывает рекламу. И, кстати, его легко установить, так что давайте приступим.

Откройте Terminal и введите следующую команду, чтобы скачать и запустить скрипт установки:

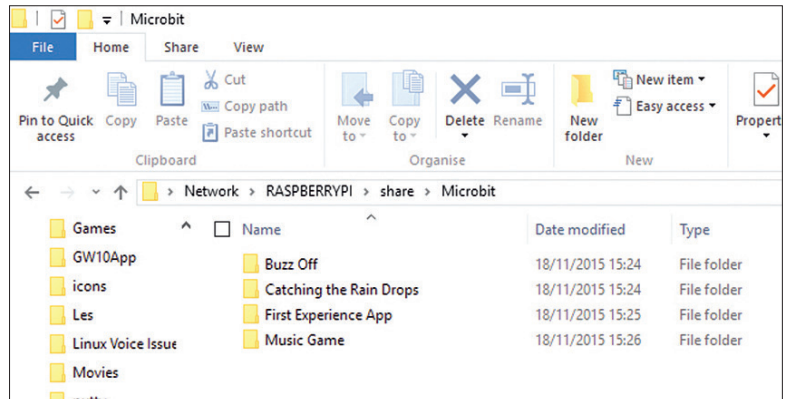
```
$ curl -sSL https://install.pi-hole.net | bash
```

Обратите внимание, что установка этим методом при использовании с неизвестными источниками несет определенный риск. В случае сомнений скачайте исходный код и проверьте его перед использованием.

Установка Pi-Hole — автоматизированный процесс, с минимальным количеством ввода. Но периодически он будет задавать вам вопросы. В один из таких моментов вас попросят задать статический IP-адрес — этого не нужно, потому что у нас уже есть статический IP-адрес, созданный для сервера *Samba* (если вы не настроили сервер *Samba*, можете задать статический IP-адрес с помощью Pi-Hole, но запишите себе этот IP-адрес).

Нас также спросят, кто является нашим вышележащим DNS-провайдером. Можно выбрать из списка: популярные варианты — Google или OpenDNS. Следуйте оставшимся этапам установки, и когда вас попросят установить web-админ, выберите On. Спустя несколько минут установка завершится, и понадобится выключить наш Pi и удалить все соединения, кроме Ethernet и кабеля питания.

Включите Pi и подождите, пока он загрузится — это занимает секунд 30. На вашем компьютере Linux/Windows перейдите в настройки сети. В Ubuntu можно нажать правой кнопкой по иконке Network [Сеть] и выбрать Edit Connections



Samba кросс-платформенный, и пользователи Windows, Mac, Linux и мобильных устройств могут получить доступ к одному центральному пункту для своих файлов.

[Редактировать соединения], нажать на интерфейс, который вы хотите отредактировать (Ethernet или Wi-Fi) и затем нажать на кнопку Edit [Редактировать]. В IPv4 Settings измените DNS-сервер, чтобы он соответствовал IP-адресу Raspberry Pi. Нажмите Save [Сохранить], Close [Закрыть], а потом нажмите на иконку Network и на Disconnect [Отключиться] для интерфейса, который вы только что настроили. Затем переподключитесь, чтобы заставить интерфейс подключиться с помощью Pi-Hole DNS сервера.

Готово! Отныне в нашей сети есть прозрачный блокировщик рекламы. Он работает со всеми устройствами, включая Android, где можно указать статический IP-адрес и DNS-сервер посредством долгого нажатия на имя Wi-Fi AP, выбора Modify Network [Изменить сеть], Advanced Options [Продвинутые опции], и ввода желаемого статического IP-адреса и DNS-сервера (наше устройство Pi-Hole).

Если вы любитель статистики, то у Pi-Hole она есть! Откройте в браузере IP-адрес сервера, а затем /admin, вы увидите интерфейс типа «приборной доски», где хранятся все записи о заблокированных запросах вашего устройства. Войдя с помощью пароля, предоставленного во время установки, вы увидите еще больше подробностей.

Итак, вот оно — два сервиса, использующих мощь новой расширенной работы в сети Raspberry Pi 3 B+, чтобы освободить наши сессии браузера от рекламы, а наши файлы — в сохранности в удобном месте. **LXF**

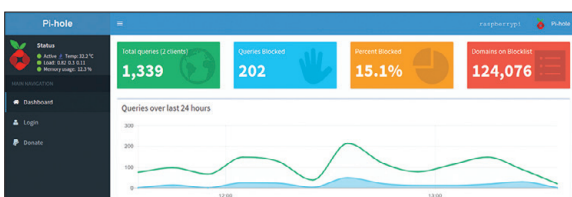
СКОРАЯ ПОМОЩЬ

У вас система Linux, которая не может смонтировать общий сетевой ресурс в файловом менеджере? Есть пакет под названием *Gigolo*, монтирующий сервера, используя SSHFS, SFTP и другие.

» АЛЬТЕРНАТИВНЫЕ МЕТОДЫ

Samba прост в настройке и работает с разными операционными системами. *Samba* также может использоваться для выдачи распределенного доступа к принтерам по сети, через сервис CUPS. CUPS легко установить, и в **LXF228** мы рассматривали, как установить CUPS на Pi Zero W и превратить старый струйный принтер в беспроводной.

NFS также предлагает преимущества пользователям Linux. Главным из них является то, что директория **Home** пользователя может обслуживаться по сети через NFS — удобно для тонких клиентских устройств; а устройства хранения информации, например, оптические диски, могут быть доступны, как если бы пользователь сидел за этим удаленным компьютером. Установка NFS проста, и имеется руководство — на <https://help.ubuntu.com/community/SettingUpNFSHowTo> — по работе в Raspbian. Если все эти разговоры о файловом сервере пробудили в вас интерес, советуем использовать Diet Pi (<http://dietpi.com>), который мы обозревали в **LXF233**. Он поставляется с автоматическим скриптом установки для различных файловых серверов, Pi-Hole и других серверов для домашней автоматизации, каталогизации медиа и торрентов. Всё это способен обеспечить скромный Raspberry Pi — согласитесь, весьма впечатляет.



После установки Pi Hole предлагает web-интерфейс с базовой статистикой, включая то, сколько рекламы было заблокировано.



УПРАВЛЕНИЕ РЕЛЕ

Изучим нагрев с логическим управлением

Шон Д. Конвэй отогревает свою машину среди канадской зимы благодаря удобному переключателю переменного тока, управляемому Raspberry Pi.



НАШ ЭКСПЕРТ

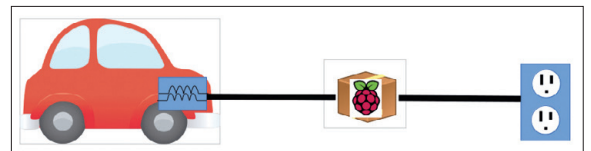
Шон Д. Конвэй днем работает специалистом по безопасности, но с заходом солнца превращается в писателя, спортивного болельщика и любителя по-мастерить что-то с Raspberry Pi.

Э тот проект появился как ответ на очень простую концепцию. Почему бы не использовать Raspberry Pi и датчик температуры для переключения на переменный ток? Это руководство подробно опишет структуру т.н. A Very Expensive Block Heater Controller [Очень дорогой обогреватель блока цилиндров] (VEBHC).

Если вы живете в климате центральной Канады, температура зимой может колебаться около -30°C , на несколько недель опускаясь ниже -40°C . Читатели из Европы по градусам Цельсия поймут, что температура низкая. Читатели, проживающие в США, сочтут температуру низкой, только если выразить ее как -22°F .

Тем, кто не жил в холодном климате, может быть трудно оценить, насколько холодно, когда температура ниже нуля. Посмотрите на эту демонстрацию замерзания: www.youtube.com/watch?v=sa1Oz1BOf1U. Это холодно!

Когда становится действительно холодно, канадские водители считают себя везунчиками, если их машины заводятся. Обогреватель блока цилиндров — это решение, которое обеспечивает некоторую защиту от мороза, мешающего машине завестись. Обогреватели блока цилиндров — это устройства обогрева на переменном токе, которые монтируются на внутренний контур



Эта схема резюмирует содержание данной статьи.

двигателя. Они разработаны, чтобы выделять тепло и повышать температуру двигателя.

Обогреватели блока цилиндров работают на переменном токе, и их непрерывная эксплуатация может обойтись недешево. Чтобы сократить рабочие расходы, обычно применяется таймер, контролирующий включение устройства. Для данной статьи мы создали такой таймер на логике скрипта Python, работающего на Raspberry Pi, который управляет твердотельным реле для направления переменного тока на обогреватель, чтобы включать и выключать его, когда надо. В процессе мы лучше поймем, почему он называется VEBHC. Вы должны осознавать опасность переменного тока от бытовой сети, и не надо подключать этот проект к сети, чтобы проверить стоящую за ним теорию на практике!

ВАМ НУЖНЫ

- Raspberry Pi Model B (Rev 2.0)
- Операционная система: Raspbian Jessie
- Приставка Wi-Fi: беспроводной адаптер Ralink Technology, Corp. RT5370
- DS3231 модуль часов 3,3 В / 5 В с высокой точностью для Raspberry Pi
- Цифровой модуль DHT22 AM2302 датчика температуры и влажности для Arduino
- Твердотельное реле SSR-50 A и 3—32 В постоянного тока, выход 50 A/250 В, 24—380 В переменного тока, с крышкой
- Одноместный короб глубиной 1-7/8 дюйма с 1/2-дюймовыми отверстиями
- Дуплексная штепсельная розетка на 15 А
- Одноместный серединный нейлоновый крепеж-основание со сдвоенной розеткой
- Двухвинтовые кабельные зажимы
- 4×2-конечных быстроразъемных соединителя Wago или Двухпозиционная 8-винтовая клеммная коробка
- Sockitbox/dribox размером 285

Подбираемся к дизайну

Перед созданием дизайн тщательно обдумывался. Устройство должно быть портативным. Оно должно иметь возможность использоваться в любых погодных условиях. Оно должно управляться дистанционно. Оно должно быть простым в создании и использовать логику для принятия решений. Оно должно использовать сенсоры, собирающие информацию об окружающей

ВНИМАНИЕ!

Электричество бытовой сети может привести к смерти или серьезным травмам. Для подключения этого проекта к сети следует обратиться к квалифицированному электрику. Выключите и отключите от питания любое оборудование перед тем, как открывать, настраивать или тестировать любую часть этого проекта. Весь риск, связанный с этим проектом, вы принимаете на себя.

среде, чтобы содействовать логике. Если время важно, то устройство должно вести точный отсчет времени — даже если теряет мощность. И, наконец, оно должно иметь возможность включать нагрузку переменного тока 120 В (в Великобритании — 240 В) 15 ампер.

Имея на руках перечень материалов, мы быстро и весьма недорого закупили всё необходимое в интернет-магазине, и зашли в местный магазин оборудования за разными мелочами. А теперь пора приниматься за сборку!

Процесс сборки

Давайте рассмотрим схему вверху справа. Питание от сети переменного тока 120 В в нижнем левом углу чертежа подключено к клеммной колодке. Три проводных вывода переменного тока подключены из клеммной колодки к одной стороне штепсельной розетки переменного тока для обеспечения питания без подключения. Черный провод, или горячая подача, разделен, и отдельное подключение осуществляется к одной стороне твердотельного реле (SSR).

С другой стороны SSR провод подключен к горячей стороне разделенной розетки. Чтобы разделить приемник, металлическая соединительная вставка на горячей стороне была удалена. Неподключенная розетка используется для питания Raspberry Pi на месте. Подключенная розетка находится там, где наш обогреватель будет получать питание 120 В переменного тока.

SSR может проводить ток до 50 А, но вряд ли достигнет этого предела, поскольку предохранитель сети переменного тока в доме вылетит при 15 А. Для улучшения рассеяния тепла SSR собран на металлической основе. Чтобы устройство не перегревалось, установите SSR на радиатор. Несколько изгибов в куске гальванизированного металла образовали тепловой радиатор и полку для монтирования SSR.

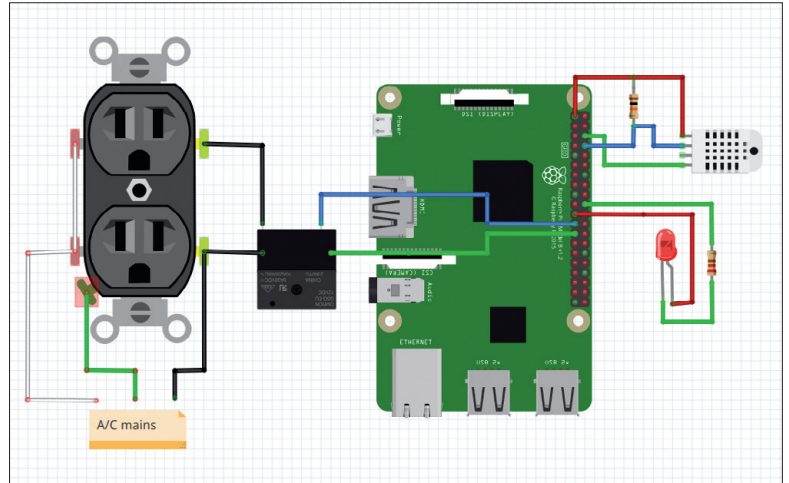
Корпус с Pi и сопутствующими компонентами является коммерчески доступным продуктом, применяемым для защиты электрических розеток поблизости от воды. Постоянные читатели видели корпус для защиты от погодных условий в **LXF232**. Он идеально подходит к задаче этого месяца, поскольку предлагает место для монтирования компонентов на деревянную доску, которую можно поместить в коробку.

Отслеживание времени

В Raspberry Pi нет системных часов. Вместо этого он генерирует и поддерживает время с помощью внутренних/программных часов. Pi с его операционной системой Raspbian может использовать сетевой протокол времени (NTP) для получения времени с другого устройства в сети с Pi, или использовать программу под названием *fake-hwclock*, чтобы периодически сохранять время и загружать его при запуске. Внутренние/программные часы Pi достаточно точно отслеживают время. Автору вспоминается блог создателя Pi, где сулилось, что за 24 часа часы Pi теряют шесть секунд.

Если мы рассчитываем, что наш PiBlock будет выполнять функции по расписанию, то нам понадобится решение, позволяющее отслеживать более точное время. Для обеспечения синхронизации часов с источником более точного времени мы могли бы использовать сеть вместе с программным решением NTP. Для получения сигнала времени NTP нужен GPS-приемник или сетевое подключение к альтернативному источнику. Поскольку PiBlock может не поддерживать надежного сетевого соединения, находясь на улице под снегом, возможно, хорошей идеей будет подкрепить отсчет времени, чтобы каждый час делалось уточнение на полсекунды.

Решением по обеспечению точного времени является включение в Pi Real Time Clock (RTC). Дополнительный модуль DS3231 — это устройство RTC размером с монету, работающее



от батареи, которое отслеживает время даже при выключенном питании. Модуль RTC содержит кристалл-осциллятор, который периодически дает Pi волшебный пинкод для обеспечения точного времени, или предоставляет точное время, когда Pi загружается. Модули DS3231 бывают разных форм и размеров. У используемого в данном проекте устройства имеется пять контактов, которые надежно фиксируются на разъеме GPIO, начиная с первого контакта.

У Pi нет базовой системы ввода-вывода (BIOS). Параметры настройки системы, которые можно вызвать из BIOS, хранятся в текстовом файле (**config.txt**). В своем любимом текстовом редакторе добавьте следующее к файлу **/boot/config.txt** для установки и настройки модуля RTC:

```
#enable RTC DS3231
dtoverlay=i2c-rtc,ds3231
```

Поскольку у нас нет планов на NTP или необходимости в сигналах *fake-hwclock*, эти программы можно удалить, введя следующее в командной строке:

```
sudo apt-get purge ntp
sudo apt-get purge fake-hwclock
```

Если у вас нет схемы, нарисуйте ее горчичей на салфетке.

СКОРАЯ ПОМОЩЬ

В сети переменного тока черный провод — горячий, белый — нейтральный, а зеленый (или голый медный провод) — заземление.

»» КАК МИНИМИЗИРОВАТЬ СВОЙ Pi

Размещение всего необходимого в конфигурации Pi можно сравнить с укладкой чемодана для поездки за границу. Самый первый шаг для Raspberry Pi — заглянуть на сайт www.raspberrypi.org. В документах в www.raspberrypi.org/documentation/installation/installing-images имеются инструкции, относящиеся к установке текущей версии операционной системы (OS).

После установки ОС есть несколько опций *raspi-config*, применимых для отладки конфигурации. Сюда входит расширение файловой системы, чтобы можно было воспользоваться преимуществами больших карт памяти, персонализация устройства посредством уточнения имени хоста и опции настройки, чтобы выяснить, на какой стороне Атлантики обитает Pi. Доступ к интерфейсу командной строки (CLI) осуществляется включением SSH и отключением GUI. Финальный штрих — смена пароля. Чтобы не потворствовать злоумышленникам, никогда не оставляйте пароль по умолчанию. На www.raspberrypi.org/documentation/configuration/raspi-config.md вы найдете подробности выполнения только что описанных задач.

Перезагрузите устройство и убедитесь, что все опции в порядке. Далее надо обновить командную строку — это следующая задача, вместе с финальной перезагрузкой, чтобы подготовить Pi для активации Wi-Fi с помощью инструкций на www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md. Советуем использовать команду `wpa_passphrase` для конфигурации. Эта опция не позволяет передавать пароли в виде простого текста.





СКОРАЯ ПОМОЩЬ

Список модулей Python можно найти на <https://docs.python.org/3/py-modindex.html>. Часть описаний взята из документации модулей. [3a] Подробности по RPi.GPIO см. на <https://sourceforge.net/projects/raspberrypython/python/> [3b], а подробности по модулю Adafruit DHT — на https://github.com/adafruit/Adafruit_Python_DHT.

Вы можете настроить правильный часовой пояс с помощью инструмента настройки или пакета *timezone*. Помните, что вам нужна только одна команда из приведенных ниже, а не обе.

```
sudo raspi-config
sudo dpkg-reconfigure tzdata
Теперь давайте установим дату:
sudo date --set '2018-01-26 18:26:00'
```

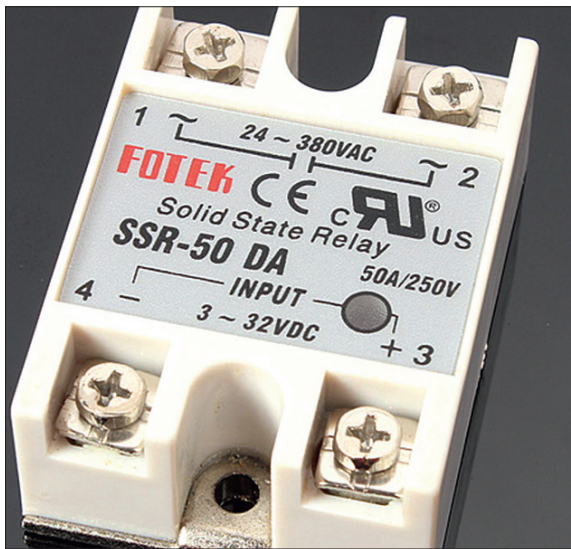
или установим пакет, который использует источник NTP, чтобы установить дату:

```
sudo apt-get install ntpdate
sudo ntpdate <замените на URL или IP-адрес источника времени>
```

Теперь вы, вероятно, удивляетесь, зачем мы раньше удалили программу NTP из Pi. *Ntpdate* — инструмент командной строки, который задает часам время из справочного источника. Это намного проще, чем устанавливать дату и время вручную. Следующий шаг — записать время и дату, полученные модулем RTC:

```
sudo hwclock -w
```

Исследуя конфигурацию, мы наткнулись на совет о том, что есть вероятность повредить время RTC скриптом оболочки или демоном NTP. Непрерывная работа PiBlock на момент написания составляла 50 дней без перезагрузки. К сожалению, мы не знаем, существует ли проблема в версии операционной системы Raspian. Если вы определите наличие проблемы, используйте текстовый



Твердотельное реле собрано на металлическом основании для рассеяния тепла.

редактор для правки параметра `HWCLOCKACCESS=NO` в файле `/etc/default/hwclock`.

Параметр точности синхронизации на модуле DS3231 указывает диапазон температур от -40°C до $+85^{\circ}\text{C}$, точность синхронизации — ± 5 PPM ($\pm 0,432$ секунды в день). Модуль RTC должен позаботиться об указанной нами погрешности в шесть секунд в день.

Структура скрипта Python

Мы в *Linux Format* публикуем статьи, которые, как мы надеемся, заинтересуют читателей, у которых может быть — или же не быть — обширная база знаний. При написании руководств важно понимать уровень знаний аудитории. Если автор предполагает у читателя некий опыт обращения с предметом, то в статью может быть меньше подробностей, чтобы задействовать ее объем по максимуму и не утомлять читателя. Если читатель — новичок, то статья, в которой всё объясняется чересчур бегло, окажется трудной для понимания и вызовет огорчение и досадливое трясение головой.

Нобелевский лауреат, физик Ричард Фейнман [Richard Feynman] понимал разницу между тем, чтобы что-то знать или знать, как это называется. В данной статье основные подробности настройки Raspberry Pi рассмотрены во врезке на стр. 89.

Давайте изучим, как открывать скрипт Python. Это шанс получить фундаментальное знание, пригодное для создания других скриптов Python в будущих проектах.

Ниже представлена среда, которую автор использует для скриптов Python. За символом `#` везде, кроме самой первой строки, следует комментарий, не являющийся частью скрипта.

```
#!/usr/bin/python
#####
# Name: ACswitch_control.py
# Описание: измерение температуры датчиками DHT22 на GPIO11
# Version 1.0
# Date: Feb 8, 2018
#####
import sys #функции манипуляции системой.
import os #удобные функции манипуляции файлами.
import datetime
import time
import RPi.GPIO as GPIO
import Adafruit_DHT #дает возможность считывать датчики DHTc Raspberry Pi
GPIO.setmode(GPIO.BOARD)
или
GPIO.setmode(GPIO.BCM)
try:
...
except KeyboardInterrupt: #if ctrl+c exit cleanly
GPIO.cleanup()
except: # улавливаются ВСЕ исключения, в т.ч. ошибки
GPIO.cleanup()
finally: #очистка GPIO при штатном выходе
GPIO.cleanup()
```

1 `#!/` — «шалаш», последовательность символов, запускающая скрипт. Следующий за этой последовательностью указатель — вызов необходимого интерпретатора языка для запуска кода внутри скрипта. В данном случае это Python. Есть разные вызовы, используемые для интерпретаторов Perl или скриптов *Bash*.

2 Это рамка для оформления титульного блока автора. Любая информация, следующая за символом `#` в скрипте — это комментарии, и они не исполняются как часть скрипта. Титульный блок обеспечивает информацию для распознавания скрипта и его

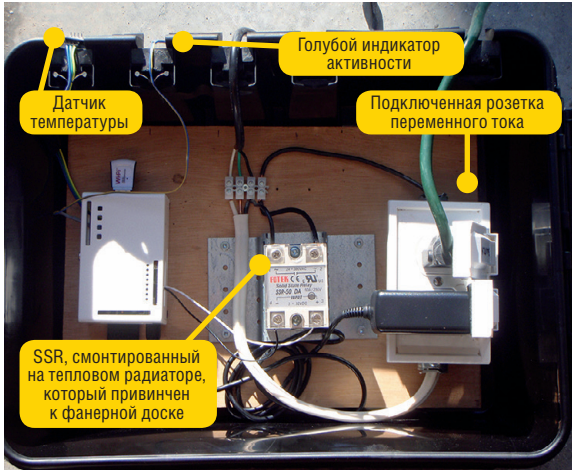
» BOARD И BCM

Опция BOARD физически устойчива независимо от вида платы Raspberry Pi. При замене платы вам не надо заново подключать провода к коннектору или менять код. Тем не менее, следует знать, какие контакты можно использовать, а какие — нет, потому что некоторые из них заняты под контакты питания и GND [Земля].

Номера BCM меняются в разных версиях модели Pi 1. Каналы BCM немного сместились в версиях 1 и 2 платы Raspberry Pi. Чтобы учесть эти изменения, всегда работайте со схемой контактов для Pi, определяя, какой номер канала относится к какому контакту платы. Номера GPIO очень удобны при взаимодействии с другими продуктами, например, от Gertboards. В этом проекте конфигурация сенсора Adafruit DHT требует справочной информации по GPIO.

Функция в модуле RPi.GPIO может использоваться для определения системы нумерации контактов (например, другим модулем Python):

```
mode = GPIO.getmode() #
Режим будет GPIO.BOARD, GPIO.BCM или None.
```

Вот что скрывается в защищенном от ненастья корпусе.

цель. Это также хорошее место для выражения благодарности, если содержание скрипта получено от других авторов.

Код Python написан, чтобы определить, как что-то сделать. Определения — это инструкции, функции и переменные, используемые в скрипте. При написании скриптов Python неплохо заимствовать код, написанный другими. Определения и операторы, хранящиеся в файле, предназначенном для интерпретатора, называются модулями. Учтите, что говоря о модулях, некоторые программисты имеют в виду «библиотеки».

Определения из модулей можно импортировать в другие модули или в главный модуль, а подборку модулей объединить в пакет. Пакет — это директория модулей Python, которые образованы сочетанием директории плюс файл `__init__.py`. И модули, и пакеты используют импорт для доступности для интерпретатора.

При использовании модуля `RPi.GPIO` надо указать одну из двух опций: `BOARD` или `BCM`. `BOARD` говорит о том, что скрипт ссылается на физические контакты на плате Raspberry Pi по номеру. `BCM` показывает, что скрипт ссылается на стандартные номера каналов Broadcom SOC, которые часто именуют номерами общего назначения ввода и вывода [general purpose input output] (GPIO).

Если вы попытаетесь остановить программу Python с помощью клавиш `Ctrl+C`, интерпретатор выдаст исключение `KeyboardInterrupt`. Когда программа заканчивается естественным образом или выдает ошибку в силу некой проблемы, она завершается, оставляя среду в условиях, которые могут быть неблагоприятными.

Одним из решений будет игнорировать предупреждения, отключив их. Это решение просто замазывает проблему, но не решает ее. Пока код не доработан так, чтобы обрабатывать исключения, любые порты, которые используются на момент получения ошибки или `KeyboardInterrupt`, останутся точно в том состоянии, в котором они были, даже после выхода из программы.

Хорошо написанная программа прибирается за собой. Исключения могут быть собраны оператором Python `catch-all tryexcept`. `RPi.GPIO` предлагает для очистки всех использованных портов встроенный метод под названием `GPIO.cleanup()`. Использование и того, и другого в коде Python перенастраивает порты, использованные в программе назад в режим ввода.

Анатомия скриптов VEBHC

Переходя к обсуждению более общих подробностей, давайте рассмотрим два скрипта Python под названием `LED_PWM_Dimmer.py` и `ACswitch_control.py`, которые поддерживают VEBHC. Дабы сэкономить место, тексты этих скриптов помещены на DVD этого месяца.

» СОВЕРШЕНСТВУЕМ ДИЗАЙН

Во время эксплуатации устройства был обнаружен просчет, с которым управались снеговая лопата и визит на место. Возникла разница в 8°C между температурой на устройстве и температурой на улице, когда контроллер оказался завален снегом почти на метр. Снег — хороший теплоизолятор, и это привело к разнице температур, из-за которой ток мог отключиться, когда он должен был быть включен. Когда устройство откопали и поместили на открытый воздух, разница в температуре исчезла.

Обнаружилось, что некоторые дополнительные функции могут улучшить дизайн. Можно было использовать средство переключения Raspberry Pi на устройстве (устройство поддерживает программное переключение). Это требует доступа к устройству через сеть. Моментальный переключатель на устройстве устранил необходимость вводить код.

Управление устройством через web-интерфейс было бы идеальным.

Пригодился бы графический инструмент для настройки расписания включений и выключений (даже в жаркие дни). Разработка воздержалась от добавления всяких излишеств для создания устройства потребительского уровня. При наличии достаточного времени умелец может добавить все функции, создающие многоплановый переключатель переменного тока. Вся инфраструктура на месте — и нужен только код, чтобы всё состоялось.

Первый скрипт, `LED_PWM_Dimmer.py`, разработан для обеспечения медленной пульсации ЖК-диода [LED] в знак того, что устройство работает. Вспомните, что все компоненты находятся в запечатанном корпусе, который подключается к розетке в стене — было бы неплохо показать пользователю, что они живы. Как печать вращающегося хэш-символа в интерфейсе командной строки показывает, что команда работает, так и скрипт создает непрерывный цикл голубого света LED, который то становится ярче, то тускнеет, но не гаснет полностью.

Чтобы выполнить задачу, скрипт использует широтно-импульсную модуляцию (ШИМ), поддерживаемую в Pi. Увеличение для LED длительности рабочей фазы (иными словами, процента цикла включения-выключения, когда LED светится) и затем ее уменьшение создает иллюзию, что LED непрерывно меняет освещение от яркого до тусклого.

Чтобы исследовать программирование Python с помощью визуального индикатора, идеальным упражнением для новичка будет использование макетной платы с цепью ШИМ с использованием Pi. Методы (например, определения и операторы внутри модуля для выполнения определенной функции) использованные для установки ШИМ, можно найти на <https://sourceforge.net/p/raspberry-gpiopython/wiki/PWM>. Найдите в Google широтно-импульсную модуляцию на Raspberry Pi, и ваша чаша знаний никогда не иссякнет.

Второй скрипт, `ACswitch_control.py`, использует текущее время и температурные данные от сенсора и пользовательскую переменную, чтобы определить, нужно ли зарядить твердотельное реле, чтобы оно обеспечивало 120 В переменного тока на включенную розетку. Если время между 3 и 8 часами утра и температура ниже -15°C, то Raspberry Pi включит SSR, при условии, что переменная перезаписи не установлена в единицу. Если переменная равна единице, то пользователю надо активировать переключатель переменного тока вручную.

Вот вам, мыслители, умельцы и программеры, создание VEBHC. На самом деле, эта разработка — просто решение по переключению переменного тока. Возможно, вы хотите, чтобы у вас включались и выключались уличные фонари или вам надо управлять электромотором. Почему бы не поместить в корпус дополнительный сенсор и не добавить собственную логику, создав собственный контроллер? **LXF**

СКОРАЯ ПОМОЩЬ

Модуль `raspberrypi-gpio-python` — не идеальный выбор для управления GPIO на Raspberry Pi в реальном времени или для управления расчетом времени в критически важной программе. Причина в том, что сложно предугадать, какую задачу выполняет Python, а также в том, что он управляется ядром в многозадачной операционной системе Linux.

Raspbian: Сборка образа

Оригинальная трактовка образа по **Евгению Голышеву** предусматривает минимально необходимую функциональность.



НАШ
ЭКСПЕРТ

Евгений Голышев всё свое рабочее время посвящает разработке микросервисов на Python и обучению своих разработчиков тому, как это правильно делать, но не перестает верить в то, что Raspberry Pi раз и навсегда изменит мир.

Тема сборки минимально функциональной версии Raspbian продолжается. Эту часть я хочу начать с обсуждения несправедливо забытого термина JeOS (читается как juice). Этот термин, который, кстати, расшифровывается как “just enough operating system”, был очень популярен лет десять назад, когда производители коммерческих дистрибутивов GNU/Linux были увлечены разработкой инструментов для индивидуальной настройки [customization] операционных систем. JeOS — это подход, согласно которому для решения какой-то конкретной задачи достаточно взять минимально возможную версию операционной системы и установить поверх нее зависимости, необходимые для запуска одного конкретного приложения, решающего эту задачу. Таким образом, решение получается самодостаточным и максимально эффективным с точки зрения производительности. В зависимости от инструмента, помогающего кастомизировать операционные системы, конечный результат был пригоден для запуска как на реальной, так и в виртуальной машине. Самым ярким из этих инструментов был SUSE Studio, который неоднократно освещался на страницах *Linux Format* (к примеру, в **LXF125** и **LXF138**). Он одним из первых начал широко использовать термин «виртуальное устройство [virtual appliance]», который означает операционную систему узкого назначения вкупе с целевым приложением, пригодную для запуска в виртуальной машине. Но времена меняются, и всё чаще для решения какой-то конкретной задачи выделяется целый Raspberry Pi или любой другой одноплатный компьютер. Терминология здесь еще недостаточно проработана, и необходимо это исправить. Назовем, к примеру, подготовленный в первой части этого руководства образ минимально функциональной версии Raspbian термином JeOSI (предлагаю произносить его как juicy). По аналогии с JeOS новый термин расшифровывается как “just enough operating system image”. С переводом будет немного сложнее. Дело в том, что словосочетание “just enough” в данном случае является наречием, а не существительным, что делает эти термины больше похожими на слоганы, чем на что-то другое. После нескольких часов размышлений я не придумал ничего лучше, чем «достаточно просто операционной системы» [можно также перевести как «ОС в обрест», — прим. ред.] для JeOS и «достаточно просто образа операционной системы» для JeOSI (как рекламный слоган популярного в 1990-х растворимого напитка Invite — «просто добавь воды»). На самом деле куда важнее, что JeOSI может сбить с толку тем, что перекликается с сетевой моделью OSI или некоммерческой организацией Open Source Initiative, название которой также часто сокращается до OSI; но более удачного термина я пока предложить не могу.

Итак, пора определиться с целями второй части учебника. Показанный в первой части подход позволяет собирать минимально функциональные образы для Raspberry Pi на основе Debian-подобных дистрибутивов. Прежде чем написать статью, я успел опробовать этот подход на Raspbian, Devuan и Ubuntu. Однако именно Raspbian использовался в качестве примера для демонстрации сборки образа на протяжении всей первой части руководства, т.к. он является «родным» для RPi. Таким образом, выбор в пользу этого дистрибутива был исключительно символическим, и нет никаких препятствий в использовании любого другого основанного на Debian дистрибутива. Экспериментируйте.

Особенность второй части учебника заключается в том, что она в основном состоит из набора не зависящих друг от друга рецептов, которые (каждый по-своему) предлагают сделать полученный в предыдущей части образ более полезным с практической точки зрения. Так что эту часть учебника можно читать как от начала до конца, так и выборочно, переходя только к тем рецептам, которые представляют наибольший интерес, без риска для понимания идущего за ними (или перед ними) материала.

Вторая часть учебника предполагает наличие образа, который у вас должен был появиться после работы с первой частью. Создайте где-нибудь две директории — **boot** и **rootfs** — и смонтируйте в них загрузочный и корневой раздел соответственно.

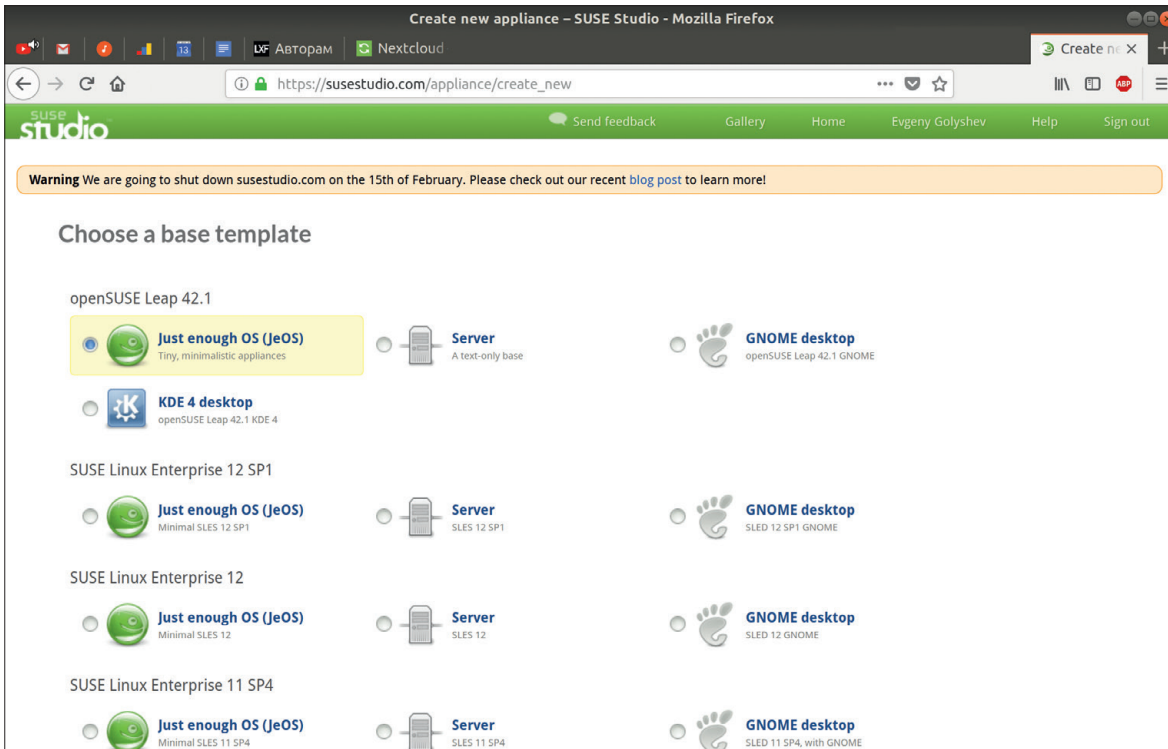
```
$ sudo mkdir /mnt/boot /mnt/rootfs
$ LOOP_DEV=$(sudo losetup --partscan --show --find raspbian-stretch.img)
$ sudo mount ${LOOP_DEV}p1 /mnt/boot
$ sudo mount ${LOOP_DEV}p2 /mnt/rootfs
```

А теперь — вперед.

Поддержка сети

Чтобы убедиться в нормальном функционировании на RPi собранного в первой части учебника образа, необходимо было подключить к устройству монитор и клавиатуру, что по очевидным причинам является не очень удобным, а в некоторых случаях — даже не всегда возможным. Поэтому в качестве первого улучшения я предлагаю добавить поддержку сети для того, чтобы можно было установить в систему SSH-сервер и подключиться к машине удаленно с любого устройства, на котором возможно запустить SSH-клиент.

Raspbian использует системный менеджер *systemd*, который, помимо всего прочего, пытается решить проблему с именован-ем сетевых интерфейсов. Таким образом, в рамках *systemd 197*,



SUSE Studio был пионером и отличным примером решения в области кастомизации операционных систем.

который вышел 7 января 2013 г., в *udev* была добавлена поддержка т. н. «предсказуемых имен сетевых интерфейсов [Predictable Network Interface Names]». И первым делом я предлагаю отключить эту поддержку. Во-первых, по иронии судьбы предсказуемые имена не всегда предсказуемы. Во-вторых, в случае RPi и его единственного проводного сетевого интерфейса нет никакой проблемы с именованием. В-третьих, я хочу, чтобы этот учебник был применим ко всем производным от Debian дистрибутивам, которые могут по тем или иным причинам не использовать *systemd*.

Чтобы отключить поддержку предсказуемых имен сетевых интерфейсов, необходимо добавить два следующие параметра к командной строке ядра, которая содержится в **cmdline.txt** на загрузочном разделе.

```
net.ifnames=0 biosdevname=0
```

Для этого добавьте в самое начало файла **cmdline.txt**, который находится на загрузочном разделе, эти два параметра, отредактировав файл вручную или воспользовавшись следующей командой:

```
$ sudo sh -c "echo 'net.ifnames=0 biosdevname=0 $(cat /mnt/cmdline.txt)' > /mnt/boot/cmdline.txt"
```

Если на данном этапе вы запишете образ на SD-карту, загрузите с нее устройство и выполните

```
$ ip link show
```

то увидите в списке всем хорошо знакомый `eth0`.

Следующим шагом необходимо разобраться с «именем хоста [hostname]». Дело в том, что *debootstrap* (см. первую часть учебника) назвал `chroot`-окружение именем системы, в которой оно собиралось, поэтому две ваши машины — RPi и та, на которой происходила сборка окружения — сейчас имеют одинаковые имена. Это необходимо исправить. Для этого придумайте новое имя для RPi и добавьте его в **/etc/hostname** и **/etc/hosts**, к примеру, следующим образом:

```
$ sudo sh -c "echo raspbian-jeosi > /mnt/rootfs/etc/hostname"
$ sudo sed -i '2i 127.0.0.1\traspbian-jeosi' /mnt/rootfs/etc/hosts
```

В данном случае в качестве имени использовалось `raspbian-jeosi`. Назовите машину с учетом ваших личных вкусов и предпочтений, но помните, что согласно RFC 952 имена хостов не должны превышать 24-х символов; в качестве символов, из которых

разрешается составлять имена — буквы латинского алфавита в верхнем и нижнем регистре, цифры, знак минуса (-) и точка (.).

В заключение необходимо установить пакеты, которые содержат DHCP-клиент, ряд вспомогательных утилит типа *ping*, всё необходимое для взаимодействия устройства с другими устройствами в сети (на базе стека протокола TCP/IP, разумеется) и SSH-сервер, ради которого всё это затевалось

```
$ sudo chroot /mnt/rootfs apt-get update
$ sudo chroot /mnt/rootfs apt-get install netbase net-tools isc-dhcp-client inetutils-ping openssh-server
```

а также отредактировать конфигурационный файл **/etc/network/interfaces**, добавив в него следующее содержимое:

```
auto lo
iface lo inet loopback
auto eth0
allow-hotplug eth0
iface eth0 inet dhcp
```

И хотя после всех этих манипуляций образ наконец имеет всё необходимое, чтобы работать в Сети, остается одна небольшая проблема с подключением к SSH-серверу. Дело в том, что OpenSSH, который достался дистрибутиву от Debian, сконфигурирован таким образом, чтобы не разрешать вход от имени `root` с парольной аутентификацией. Решить эту проблему можно одним из двух способов: либо измените в **/etc/ssh/sshd_config** текущее значение `PermitRootLogin` на `yes`, включив тем самым возможность входа как `root` по паролю (что настоятельно не рекомендуется), либо перейдите к следующему разделу и заведите еще одного пользователя в системе, на которого не будет накладываться это ограничение.

Пользователи

На данный момент в системе есть только один пользователь — `root`. Этого было достаточно, чтобы, не отвлекаясь на лишние детали, убедиться, что новоиспеченная система находится в рабочем состоянии. Теперь настало время обратить внимание на опущенные ранее подробности.

В настоящее время разработчики дистрибутивов делятся на два лагеря, когда речь заходит о пользователе `root` (напомню,



Получив дополнительные права через `sudo`, не забывайте, что с большой властью приходит большая ответственность.

```
eugulixes@ubuntu-mac: ~
File Edit View Search Terminal Help
eugulixes@ubuntu-mac:~$ sudo test

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for eugulixes:
```

что по отношению к нему часто используется термин «суперпользователь»). Более консервативные следуют практике наличия по крайней мере двух учетных записей в системе: `root`'а для административных задач и непривилегированного пользователя для всех остальных. Этот канонический подход дошел до нас в почти неизменном виде с самых первых версий UNIX; но он не лишен недостатков. На мой взгляд, наиболее ярким из них является необходимость постоянного контроля за тем, чтобы сессия `root`'а была вовремя закрыта. Таким образом, всегда есть риск забыть и продолжить решение своих повседневных задач от лица суперпользователя, что может привести к серьезным негативным последствиям — ведь безграничные возможности, скрытые в учетной записи `root`'а, обладают поистине разрушительной силой, которую не каждый способен держать под контролем.

Debian является примером одного из консервативных с этой точки зрения дистрибутивов. Ubuntu, в свою очередь, это один из первых дистрибутивов, который взял на вооружение подход, при котором все административные задачи решаются посредством `sudo`. Суть этого подхода заключается в том, чтобы

- не устанавливать пароль для суперпользователя, оставляя его учетную запись заблокированной, или заблокировать учетную запись явно, указав вместо пароля (или перед паролем) восклицательный знак (!);

- установить программу `sudo`;
- добавить первого пользователя (т. е. того, который был создан во время установки системы) в группу `sudo`, дав ему тем самым неограниченные права при выполнении любых операций в системе.

С одной стороны, такая практика противоречит философии программы — дать каждому пользователю как можно меньше привилегий, но так, чтобы их было достаточно для решения поставленных перед этими пользователями задач; а с другой — помогает решить проблему с закрытием сессии `root`.

В официальном образе Raspbian используется именно этот подход. Какой подход будет использоваться в вашем образе — решать вам, но для начала необходимо завести в системе еще одного пользователя, от имени которого будет осуществляться вход, в том числе и через SSH.

Итак, чтобы создать пользователя с именем `pi`, выполните

```
$ sudo chroot /mnt/rootfs useradd -m -s /bin/bash pi
```

Заметьте, что `useradd` и другие смежные с ней программы, в том виде, в котором они используются в этом разделе, специфичны для всех систем, в которых используется `shadow` (<https://github.com/shadow-maint/shadow>) для управления

пользователями и группами. В отличие от `coreutils`, интерфейс программ (и даже название самих программ) для управления пользователями и группами сильно варьируется от реализации к реализации, поэтому всё описанное здесь может отличаться от того, что принято в других Unix-подобных операционных системах и даже других дистрибутивах GNU/Linux.

Опция `-m` позволяет указать, что в `/home` необходимо создать домашнюю директорию пользователя с его именем, а опция `-s` говорит, что оболочкой по умолчанию должна стать `Bash`. Можно было бы посредством опции `-p` указать пароль, но это небезопасно. Во-первых, вышеприведенная команда останется в истории, к которой может получить доступ администратор или злоумышленник, если пользователь, от имени которого она выполнялась, будет скомпрометирован. Во-вторых, процесс создания пользователя будет фигурировать в списке процессов, из которого администратор может получить все переданные `useradd` параметры. В-третьих, пароль на экране может быть просто подсмотрен из-за плеча. Даже если эта и другие приведенные в данной статье команды выполняются на персональной машине в пустом и запертом помещении, я всё равно предлагаю пойти правильным путем и установить пароль пользователя отдельно. В конце концов, чтобы стать настоящим хакером, нужно быть немного шизофреником.

```
$ sudo chroot /mnt/rootfs passwd pi
```

На данном этапе образ имеет всё необходимое, чтобы считаться приверженцем консервативного подхода. Чтобы заставить его следовать подходу, при котором все административные задачи решаются через `sudo`, сначала установим саму программу, а затем добавим пользователя в соответствующую группу.

```
$ sudo chroot /mnt/rootfs apt-get update
```

```
$ sudo chroot /mnt/rootfs apt-get install sudo
```

```
$ sudo chroot /mnt/rootfs usermod -aG sudo pi
```

Теперь пользователь `pi` может временно позаимствовать права суперпользователя для тех вещей, которые выполняются через `sudo`.

Финальный штрих — блокировка учетной записи `root`. Для этого выполните

```
$ sudo chroot /mnt/rootfs passwd -l root
```

Данная команда поставит восклицательный знак (!) перед хэшем пароля, и это обозначит, что учетная запись заблокирована. Загляните в «теневой файл паролей [`shadowed password file`]» `/etc/shadow`, чтобы в этом убедиться.

На этом тему можно было бы считать закрытой, но остается еще одна вещь, которая не дает мне покоя. Дело в том, что за полтора десятка лет существования Ubuntu успело подрасти

целое поколение линуксоидов, которые убеждены, что `sudo` — это не более чем просто способ временного получения прав суперпользователя, и если оставить всё как есть, то наш учебник только поддержит это распространенное заблуждение. Я хочу внести свой посильный вклад в то, чтобы исправить сложившуюся ситуацию. Для этого я предлагаю решить одну простую, но очень актуальную для RPi задачу, суть которой заключается в том, чтобы дать пользователю возможность выключать или перезагружать свое устройство через Android-приложение нажатием одной кнопки. Таким образом, потребуется посредством `sudo` дать пользователю право выполнять команды `systemctl poweroff` и `systemctl reboot`. Более того, необходимо не только дать пользователю право выполнения этих команд, но и при этом не спрашивать его пароль. Дело в том, что Android-приложение SSH Button, которое я предлагаю использовать для создания кнопок выключения и перезагрузки устройства, как и любое подобное приложение, очень примитивно. Оно подключается к SSH-серверу, выполняет указанную программу и анализирует код завершения [exit status]. Если в результате своей работы программа переходит в режим, в котором она ожидает каких-то действий от пользователя (к примеру, ввода пароля), то SSH Button считает выполнение программы неудачным. Всё это создает идеальные условия для того, чтобы копнуть чуть глубже возможности, предлагаемые `sudo`, чем я и предлагаю заняться прямо сейчас.

Правила, согласно которым `sudo` принимает решение, кто и что в праве делать, находятся в `/etc/sudoers`. Однако, вместо того, чтобы редактировать этот файл напрямую, я предлагаю добавить новое правило в виде отдельного файла через подключаемую директорию `/etc/sudoers.d`. Создайте файл, к примеру, с именем `shutdown` со следующим содержимым.

```
pi raspbian-jeosi =NOPASSWD: /usr/bin/systemctl halt,/usr/bin/systemctl reboot
```

Все файлы в этой директории должны иметь права 0440 (`-r--r-----`).

```
$ sudo chmod 0440 /mnt/rootfs/etc/sudoers.d/shutdown
```

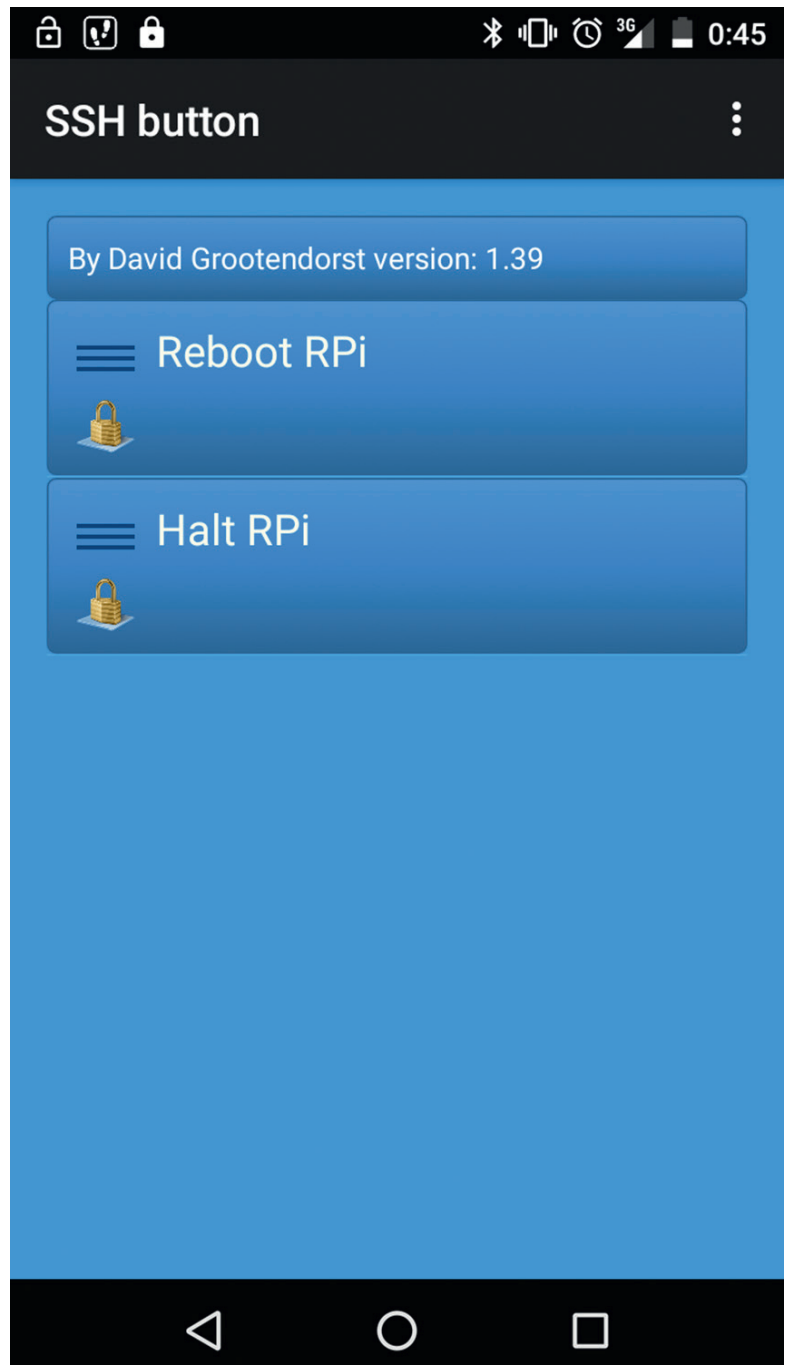
Теперь установите SSH Button на любое устройство под управлением Android. Затем создайте в нем кнопку и укажите в ее настройках адрес вашего RPi, логин и пароль пользователя `pi` и одну из разрешенных этому пользователю команд — `systemctl poweroff` или `systemctl reboot`.

Этот подход особенно удобен, когда вам нужно подключиться к RPi только для того, чтобы корректно завершить все процессы на нем перед его выключением. Читайте следующий раздел, в котором пойдет речь о том, как сделать корневую файловую систему доступной только на чтение и, таким образом, избавиться от необходимости т.н. корректного завершения работы.

Доступный только на чтение корень

Продвинутые владельцы одноплатников, и тем более те, кто создает на их основе продукты, предпочитают (очевидно, если это не противоречит поставленной задаче) делать корневую файловую систему доступной только на чтение. В таком случае можно не переживать за целостность данных из-за внезапного отключения питания.

Доступный только на чтение корень — это умышленное ограничение функциональности системы для повышения ее живучести. Но основная проблема здесь не в том, как этого добиться, а в том, как минимизировать последствия от этого ограничения: ведь будет очень обидно, если демон, предназначенный для решения той самой единственной задачи, потеряет способность вести журнал... Тому, как сделать корень доступным только на чтение и не терять логи в этих условиях, посвящена большая часть этого раздела. (В силу того, что материал руководства выходит далеко за пределы журналирования, разговор на эту тему



не будет исчерпывающим, поэтому я настоятельно рекомендую обратиться к статье «Записки демонов» из **LXF91** за более детальными подробностями по этому вопросу.)

Традиционно в Unix-подобных операционных системах журналированием занимается демон `syslogd`. В Debian, к примеру, используется одна из его реализаций под названием `rsyslog`. Когда сообщение доходит до `syslogd`, возможен один из пяти вариантов развития событий:

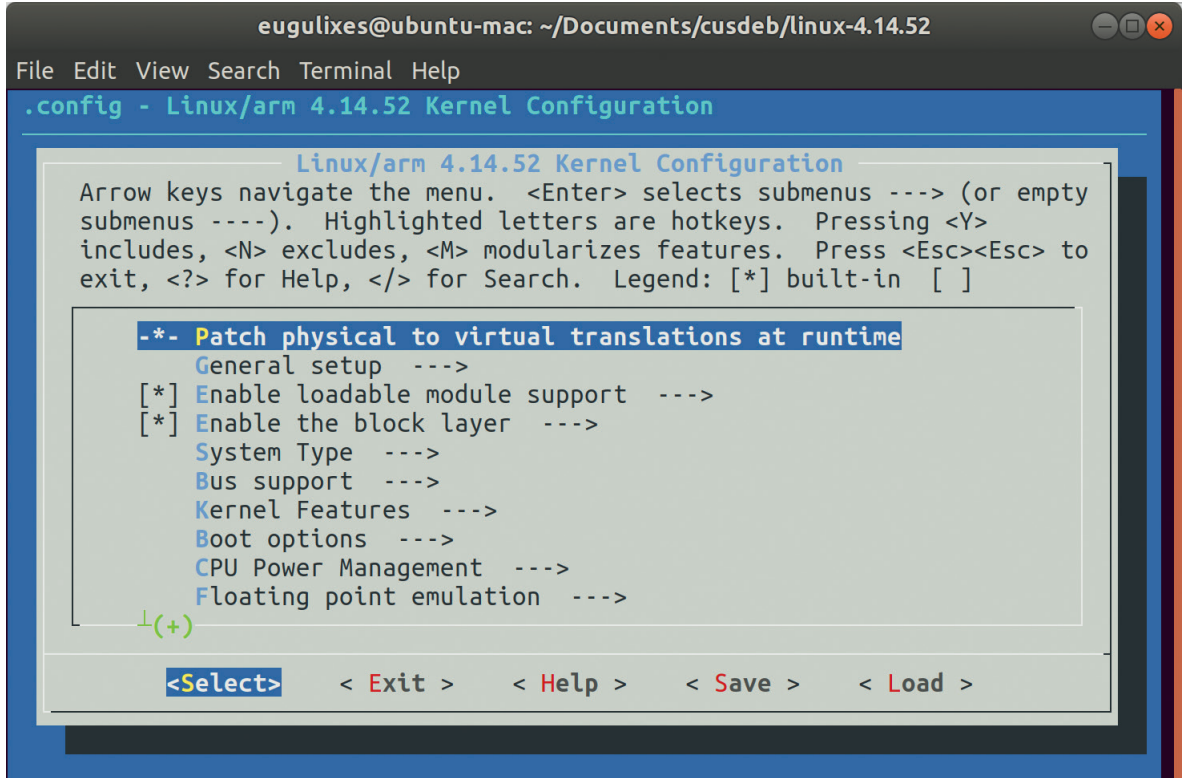
- сообщение может быть добавлено в файл;
- сообщение может быть выдано на терминал любого указанного пользователя;
- сообщение может быть записано в FIFO (именованный канал);
- сообщение может быть перенаправлено `syslogd`, находящемуся на другой машине;
- сообщение может быть проигнорировано.

Самый распространенный первый вариант не подходит, т.к. корневая файловая система доступна только на чтение. Можно,

Удобный способ выполнить любую команду на RPi с мобильного устройства.



menuconfig поможет сконфигурировать ядро для решения конкретной задачи.



конечно, передавать логи на другую машину, но в некоторых случаях наличие еще одной машины может оказаться избыточным. К счастью, другая реализация *syslogd*, разработанная в рамках проекта Busybox, поддерживает использование так называемого «кольцевого буфера [circular buffer]» для хранения логов. Таким образом, создается иллюзия того, что логи пишутся в обычном режиме так, как если бы корневая файловая система не была доступна только на чтение.

В Raspbian (и других дистрибутивах, основанных на Debian) реализация *syslogd* от проекта Busybox находится в пакете *busybox-syslogd*.

```
$ sudo chroot /mnt/rootfs apt-get install busybox-syslogd
```

Если в системе уже был установлен *rsyslog*, то установка *busybox-syslogd* приведет к его удалению, т.к. *busybox-syslogd* возьмет на себя функции *rsyslog*.

Теперь логи подавляющего большинства демонов в вашей системе будут сохраняться в памяти. *busybox-syslogd* сконфигурирован в Raspbian таким образом, чтобы использовать для этих целей буфер размером 64 КБ. Однако, если произойдет внезапное отключение питания, то все логи вылетят в трубу. Конечно, самым надежным во всех случаях подходом является хранение логов на отдельной машине, и *busybox-syslogd* это тоже умеет.

Несмотря на то, что подавляющее большинство демонов используют *syslogd* для журналирования, есть демоны, которые занимаются этим вопросом самостоятельно. К примеру, *Nginx* ведет два типа журналов: журнал доступа [access log] и журнал ошибок [error log], и по умолчанию не пользуется услугами *syslogd* для этих целей. Тем не менее, модуль *Nginx ngx_http_log_module*, который занимается журналированием, поддерживает использование *syslogd*. В Debian и его производных *Nginx* сконфигурирован так, чтобы журнал доступа сохранялся в `/var/log/nginx/access.log`, а журнал ошибок — в `/var/log/nginx/error.log`. Очевидно, что в файловой системе, доступной только на чтение, *Nginx* не сможет писать в `access.log` и `error.log`, поэтому необходимо попросить web-сервер регистрировать события через *syslogd*. Для этого отредактируйте `/etc/nginx/nginx.conf` так, чтобы директивы `access_log` и `error_log` выглядели следующим образом:

```
access_log syslog:server=unix:/dev/log,nohostname;
```

```
error_log syslog:server=unix:/dev/log,nohostname;
```

И в заключение, необходимо сделать корневой раздел доступным только на чтение. Для этого добавьте параметр `ro` к командной строке ядра, которая находится в файле `cmdline.txt` на загрузочном разделе.

Сборка ядра

Сейчас система основана на Raspbian'овском ядре, которое латается и пакетируется самими разработчиками одноплатников, поэтому оно, как ничто другое, отлично поддерживает RPi. Пакет *raspberrypi-kernel*, установка которого обсуждалась в первой части этого руководства, содержит ядро, предназначенное для решения широкого круга задач, что открывает огромные возможности для оптимизаций. Таким образом, в данном разделе речь пойдет о том, как самостоятельно собрать ядро — для того, чтобы получить возможность подогнать его под решение конкретной задачи.

Теперь осталось определиться, у кого мы возьмем ядро, которое будем здесь собирать: у Линуса, разработчиков RPi или у кого-то еще. Предлагаю собирать именно «ванильное» ядро, т.к. этот подход является наиболее дистрибутивнезависимым. Базовая поддержка RPi появилась в upstream'e достаточно давно. К примеру, RPi 2 поддерживается Linux'ом, начиная с версии 4.5, которая вышла 13-го марта 2016 г., а RPi 3 — начиная с версии 4.8, которая вышла 2 октября того же года. Безусловно, RPi поддерживается «ванильным», т.е. стандартным, ядром не настолько хорошо, как Raspbian'овским, но для решения некоторых задач этим можно пренебречь.

В качестве примера я возьму Linux 4.14 — последний на момент написания статьи выпуск с длительным сроком поддержки. Так, этот раздел не потеряет своей актуальности ни на грамм, пока не закончится жизненный цикл выпуска 4.14, которое при участии Google будет сопровождаться до 2023 г. Тем не менее, я призываю вас брать самую последнюю на момент чтения этого руководства версию ядра и применять к ней описанные здесь рецепты.

А теперь — к делу. Загрузите с kernel.org архив с интересующей вас версией ядра Linux. К примеру, им оказался **linux-4.14.52.tar.xz**.

Распакуйте его.

```
$ tar xJvf linux-4.14.52.tar.xz
```

Затем установите необходимые для конфигурации и сборки ядра пакеты (в Debian и производных от него это *build-essential*, *libncurses5-dev* и *gcc-arm-linux-gnueabi*). После этого выполните следующие команды, чтобы приступить к конфигурации целевого ядра, ради которой мы здесь все собрались.

```
$ ARCH=arm make bcm2835_defconfig
```

```
$ ARCH=arm make menuconfig
```

На этом этапе, как правило, из ядра выкидывается всё, что не способствует выполнению поставленной задачи, и добавляется всё, чего не хватает. Руководство не предлагает в качестве примера какую-то конкретную задачу, поэтому здесь всё зависит от вас. Я настоятельно рекомендую не закликиваться на этом шаге, двинуться дальше, а потом еще раз вернуться к этому разделу после просмотра доклада “*Tuning Linux For Embedded Systems: When Less is More* [Настройка Linux для встраиваемых систем: Когда больше — это меньше]” Даррена Гарта [Darren Hart], где он ставит перед собой цель собрать минимально возможное ядро для встраиваемого устройства.

После того как с конфигурацией будет покончено, можно начинать сборку:

```
$ ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-hf- chrt -i 0 make -j4 deb-pkg
```

В результате получится несколько Deb-пакетов. Сейчас нас интересует *linux-image*, полное название у которого в моем случае — **linux-image-4.14.52-4.14.52-1_armhf.deb**. Установите его, предварительно удалив пакет с Raspbian’овским ядром:

```
$ sudo chroot /mnt/rootfs apt-get purge raspberrypi-kernel
```

```
$ sudo cp linux-image-4.14.52-4.14.52-1_armhf.deb /mnt/rootfs
```

```
$ sudo chroot /mnt/rootfs dpkg -i linux-image-4.14.52-4.14.52-1_armhf.deb
```

```
$ sudo rm /mnt/rootfs/linux-image-4.14.52-4.14.52-1_armhf.deb
```

Затем удалите с загрузочного раздела всё, что относится к Raspbian’овскому ядру и загрузите на него новое ядро и dtb-файлы:

```
$ sudo rm /mnt/boot/cmdline.txt,bcm283*,kernel7.img}
```

```
$ sudo cp /mnt/rootfs/boot/vmlinuz-4.14.52 /mnt/boot/zImage
```

```
$ sudo cp /mnt/rootfs/usr/lib/linux-image-4.14.52/bcm283* /mnt/boot
```

Для загрузки «ванильного» ядра потребуется загрузчик, в качестве которого я предлагаю использовать *Das U-Boot*. Последней стабильной версией *U-Boot* на момент написания статьи является 2018.05. Загрузите архив с исходниками *U-Boot*, укажите файл конфигурации, которая соответствует вашему устройству, и запустите сборку.

Что касается файла конфигурации, то в моем случае это **rpi_3_32b_defconfig**, т.к. я хочу запустить на Raspberry Pi 3 Model B собранное под ARMv7 ядро Linux, поскольку мы имеем дело с 32-битной пакетной базой Raspbian. Подсмотрите в директории **configs** все возможные варианты.

```
$ curl -O ftp://ftp.denx.de/pub/u-boot/u-boot-2018.05.tar.bz2
```

```
$ tar xjvf u-boot-2018.05.tar.bz2
```

```
$ cd u-boot-2018.05
```

```
$ ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-hf- chrt -i 0 make rpi_3_32b_defconfig
```

```
$ ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-hf- chrt -i 0 make -j4
```

Теперь подготовьте файл **boot.scr** со следующим содержанием:

```
mmc dev 0
```

```
setenv fdtfile bcm2837-rpi-3-b.dtb
```

```
setenv bootargs earlyprintk console=tty0 console=ttyAMA0
root=/dev/mmcblk0p2 rootwait init=/bin/systemd
fatload mmc 0:1 ${kernel_addr_r} zImage
fatload mmc 0:1 ${fdt_addr_r} ${fdtfile}
bootz ${kernel_addr_r} - ${fdt_addr_r}
```

Этот файл содержит список команд, которые *U-Boot* должен выполнить. Такой способ конфигурации *U-Boot* делает загрузчик очень гибким.

Есть три места, на которые в этом пункте стоит обратить пристальное внимание.

- 2-я строка содержит имя DTB-файла, соответствующего устройству, на котором планируется загрузка ядра. Указанный в этой строке файл, в числе прочих DTB-файлов, был получен при сборке ядра и уже находится на корневом и загрузочном разделах. Укажите здесь DTB-файл, соответствующий вашему устройству.

- 3-я строка содержит командную строку ядра.

- 4-я строка содержит имя двоичного файла ядра (*zImage*).

Теперь необходимо скомпилировать **boot.scr** в **boot.scr.uimg**.

Для этого потребуется программа *mkimage*, которую можно найти в пакете *u-boot-tools* во всех Debian-подобных дистрибутивах или в *u-boot-tools* в Fedora.

```
$ mkimage -A arm -O linux -T script -C none -n boot.scr -d boot.scr boot.scr.uimg
```

Наконец, нужно перекинуть двоичный файл загрузчика, который был получен после сборки *U-Boot*, и **boot.scr.uimg** на загрузочный раздел:

```
$ sudo u-boot.bin /mnt/boot/kernel7.img
```

```
$ sudo boot.scr.uimg /mnt/boot
```

Очистка образа

Неотъемлемой частью подготовки образа для RPi и других одноплатников является его очистка. Если образ решает какую-то пространственную задачу, то он заслуживает того, чтобы им пользовались как можно большее количество людей. Но крайне непрофессионально распространять образ, в котором остаются следы работы над ним. Это в первую очередь относится к кэшам двоичных пакетов и индексным файлам.

Существует идиома, которая решает эту проблему. Она стала популярной за счет повсеместного использования в Dockerfile’ах, т.к. Docker-образы, как и образы для одноплатников, тоже нуждаются в очистке от «строительного мусора». Выглядит эта идиома следующим образом.

```
$ sudo apt-get clean
```

```
$ sudo rm -rf /var/lib/apt/lists/*
```

Первая команда удаляет кэшированные двоичные пакеты, а вторая — индексные файлы. В нашем случае эти команды будут выглядеть так:

```
$ sudo chroot /mnt/rootfs apt-get clean
```

```
$ sudo rm -rf /mnt/rootfs/var/lib/apt/lists/*
```

Заключение

Несмотря на то, что есть достаточно большое количество скриптов, автоматизирующих весь описанный в этом руководстве процесс, я убежден в большой пользе понимания, что у них под капотом. Во-первых, эти скрипты нужны в контрибуторах — и где как не здесь их выращивать. Во-вторых, спускаясь на уровень сборки своей системы, хоть и на основе двоичных пакетов, есть возможность лучше понять GNU/Linux, а где как не здесь этим заниматься. Я старался выжимать максимум из каждого раздела этой части руководства, выстраивая как можно более длинные цепочки из смежных тем для того, чтобы сделать материал как можно более энциклопедическим. Теперь ваша очередь — сдуйте пыль со своего одноплатника и выжмите из него максимум. Удачного хакинга. **LXF**

Ответы



Джонни Бидвелл

Пытается уладить ваши разборы с Туксом.

Вас жгут вопросы по открытому коду или ядру? Каким бы ни был ваш уровень, присылайте их на lxf.answers@futurenet.com, чтобы получить решения и предложения от знаменитого **д-ра Бидвелла**.

В Какие новости?

Нынче трудно найти хорошую читалку RSS. *Liferea* работает, но выглядит слегка устаревшей. Похоже, у каждого второго клиента свои странности и причуды. Есть масса размещенных сервисов, многие из которых хуже когда-то популярного, но уже не работающего сервиса Google Reader. *Feedly* популярен, но если я буду использовать размещенный сервер, я уж точно предпочту тот, который сам и размещаю.

Недавно я обновил мой экземпляр Nextcloud до 13.0 и решил попробовать его приложение News. Оно работает очень хорошо, я могу использовать приложение Android для чтения новостей на телефоне и использовать *FeedReader* на моем рабочем столе. Однако я вижу эту странную ошибку в самом приложении News:

```
Non UTF-8 charset for MySQL/MariaDB database detected! Learn how to convert your database
```

```
to utf8mb4 (make a backup beforehand).  
[Обнаружен отличающийся от UTF-8 набор символов для базы данных MySQL/MariaDB! Узнайте, как конвертировать вашу базу данных в utf8mb4 (сначала сделайте резервную копию)]
```

Ошибка дает ссылку на пост StackExchange с весьма сложными инструкциями для конвертирования вручную таблиц SQL. Это выглядит не слишком хорошим способом работы, поэтому я умоляю гуря Nextcloud в редакции LXF подсказать мне более удобное решение. Зачем мне вообще нужен этот странный набор символов?

Бонни Арчвелл [Bonnie Archwell]

О Нам тоже нравится приложение Nextcloud News, и мы тоже встречали эту ошибку. Мы последовали инструкциям в указанном посте StackExchange, но безрезультатно. Обсуждаемая проблема — четырехбайтный набор символов

UTF-8, который отвечает за эмодзи. Картинка может заменить тысячи слов, а элементам пространства кодов, представляющим эмодзи, нужно всего четыре байта. Если Вы в эмодзи не нуждаетесь и Вам не мешает эта ошибка, просто не обращайтесь на нее внимания. Но если Вы хотите видеть в своей новостной ленте опрокинутые лица и прочие гримасы, читайте дальше. Сначала сделайте, пожалуйста, резервные копии своих данных. Для смены настройки набора символов можете использовать *oss*, встроенную утилиту Next/Owncloud, и затем установите свой экземпляр Nextcloud в режим Maintenance, чтобы обновить таблицы.

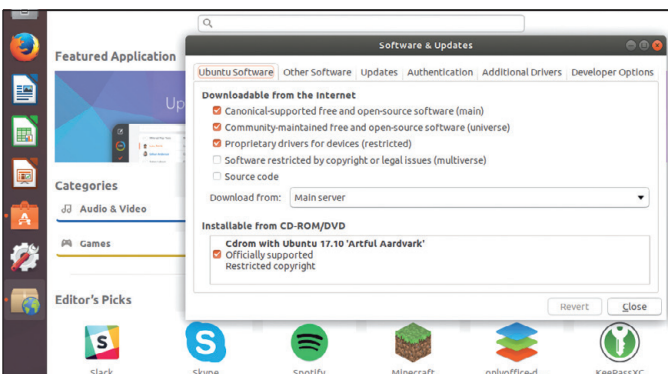
Однако, прежде чем это сделать, надо внести кое-какие поправки в базу данных. Более широкий набор символов требует более длинного префикса ключа, иначе Вы получите устрашающую ошибку. Поэтому добавьте следующие настройки в раздел InnoDB конфигурации Вашей *MySQL/MariaDB* (в Debian файл настройки сервера по умолчанию — `/etc/mysql/mariadb.conf.d/50-server.cnf`):

```
innodb_large_prefix=on  
innodb_file_format=barracuda  
innodb_file_per_table=true
```

Теперь перезапустите SQL-сервер — `systemctl restart mysqld` — и запустите *oss* таким образом:

```
$ cd /var/www/nextcloud  
$ sudo -u www-data php occ config:system:set  
mysql.utf8mb4 --type boolean --value="true"  
$ sudo -u www-data php occ maintenance:repair
```

Это должно сработать. Если режим поддержки еще активен (Nextcloud сообщит Вам об этом, когда Вы попытаетесь войти), то нормальный режим можно восстановить, заменив аргумент `maintenance` в последней команде на `maintenance:mode --off`.



Возможно, традиционалисты предпочли бы просматривать программы через *Synaptic* или нечто подобное, а не с помощью приложения *Ubuntu Software*.

»» СЕЗАМ, ОТКРОЙ (КОД)

В О материале *Сравнения в LXF233* я думаю, что в статье нет главного: мне бы нужно решение для хранения паролей, хорошо работающее на Android и настольных Windows и Linux. Поскольку некоторые из этих кандидатов используют облачное хранилище, я так понимаю, целью разработки является многоплатформенная поддержка. Возможно, пока что нет решений с открытым кодом, но вы дали мне отправную точку. Может, у вас найдется место для продолжения этой темы? Читаю с удовольствием.
Колм, по электронной почте

О Отлично подмечено. Я как раз выполнил крупномасштабную смену паролей, потратив Очень Много Времени — в основном на поиск сайтов, где у меня сохранились активные учетки. Когда этот список был готов, еще уйма времени ушла на поиск качественного менеджера паролей FLOSS, и я остановился на *Padlock*. Вы можете надежно хранить свои данные с помощью облачного сервиса *Padlock*, на его облаке или у себя. Также можно воспользоваться штучным решением вроде *Duplicity* и организовать себе безопасное хранилище.

В Я предпочитаю ремикс

Я слежу за новостями о ремиксах. Мне удалось установить стандартную опцию Ubuntu. Это рабочий стол *Gnome* по умолчанию, который сейчас работает отлично. До этого я использовал рабочий стол *Unity* в Ubuntu, так что для меня всё это вновь. Что мне теперь делать? В списке приложений не предлагается других рабочих столов! И я, похоже, не могу осуществлять поиск из приложения *Ubuntu Software*!

Что мне бы хотелось сделать в *Gnome*, так это установить шахматную программу *Windows (Fritz 8)* на Ubuntu 17.10. Я установил *Play-OnLinux* с помощью терминала. Теперь оно есть

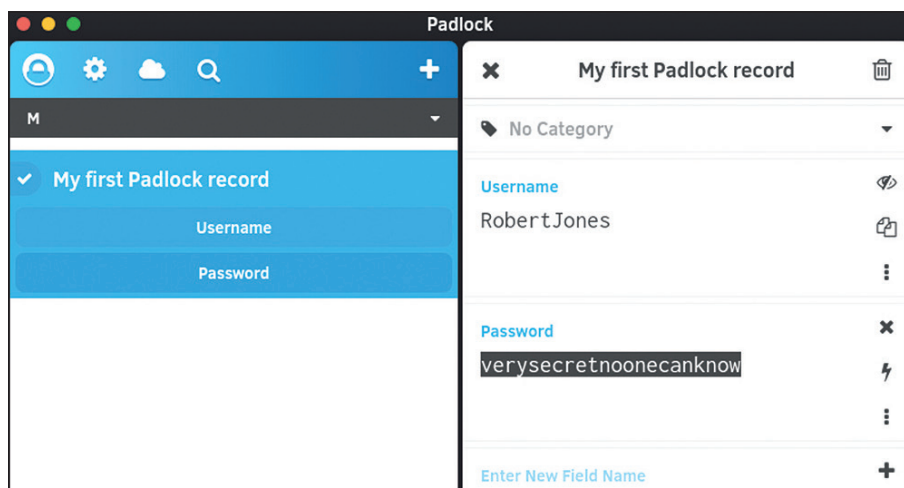
на рабочем столе и я могу открывать его, когда мне это надо. Однако у меня не получается установить его, потому что оно выдает ошибку: `-error-wine has crashed`. Это неожиданно, потому что я устанавливал *Fritz 8* на другие старые компьютеры с Ubuntu 12.04 и 14.04. *WINE* вроде и установлен, но я не могу его запустить.

Билл, по электронной почте

О Недолго осталось дожидаться следующего Ubuntu LTS — он должен быть мил. Я рад, что Вам пока что нравится 17.10. Мы считаем, что в Canonical проделали отличную работу со своим рабочим столом Gnome. Он приобрел немалое сходство с Unity, но не употребляет такого множества агрессивных дополнений.

Однако Gnome подойдет не всем, и именно поэтому мы показали некоторые альтернативы (а также обалденные навыки г-на Ботвика в реинжиниринге) в нашем ремиксе в **LXF231**. А что Вам теперь надо делать, зависит от вас и от ваших планов. Репозитории по умолчанию (которые должны работать «из коробки») дадут Вам доступ к тысячам отличных инструментов. У инструмента Software на самом деле имеется утилита поиска, но если ни одна программа не отображается, будем надеяться, это можно исправить с помощью инструмента *Software & Updates*.

Возможности установить ремикс напрямую нет, он чисто для демонстрации. Но легко добавить любой или все рабочие столы, там



Padlock доступен в качестве мобильного приложения или на рабочем столе как AppImage. Его написанный на Go облачный сервер Padlock Cloud бесплатный.

имеющиеся. Большинству не нужны они все, поскольку в итоге получается несколько приложений в разном стиле, которые делают одно и то же. Используйте `sudo apt install xubuntu-desktop` для добавления, например, рабочего стола *Xfce* (и всех приложений по умолчанию из Xubuntu). Если Вам просто нужен сам рабочий стол, тогда используйте `sudo apt install xfce4`.

Хотя активная разработка Unity прекращена, он всё же имеется в репозиториях, и он там останется и будет получать отладки в 18.04. Его

можно добавить с помощью `sudo apt install unity`, но мы бы рекомендовали дать шанс Gnome или приискать новый любимый рабочий стол.

Заставить работать программы Windows — дело непростое. *WINE* и вправду будет установлен, если вы использовали PlayOnLinux, потому что именно его применяет PoL за кулисами, но поначалу работать с ним может быть страшно.

Попутно замечу, что *Fritz 8* уже не юн, и сейчас есть масса превосходных шахматных программ с открытым кодом, достойных внимания [Ред.: — Притворюсь, что я этого не читал]. Вы также можете поиграть в более современную версию *Fritz* прямо в браузере, на <https://fritz.chessbase.com>. **LXF**

» ТАЙНЫЙ ОРДЕН РАЗРАБОТЧИКОВ

В Меня всегда интересовало, как выглядит среда разработчика Linux. Они по-прежнему предпочитают для разработки настольные ПК? Есть ли у них отдельный компьютер для разработки и для личного пользования (из-за обновлений библиотек) или они используют виртуальные машины? Предпочитают ли они какой-то определенный рабочий стол?

Спасибо за то, что уделите время.

Пег Расселл [Peg Russell], по электронной почте

О Похоже, бытует мнение, что разработчики — некий особый вид, который имеет дело с тайными эзотерическими знаками, манадами и горячими клавишами Vim. Может, это и правда, но мне больше нравится думать, что научиться программировать может каждый, и, следовательно, каждый может стать разработчиком.

Впрочем, разработчики (как и прочие существа) имеют свои предпочтения, но для многих избранным орудием является MacBook (если не верите, зайдите на любую конференцию Linux). Конечно, многие из них работают на Linux, и одни предпочитают графические рабочие столы, а другие — более минималистские среды. Многие из тех, с которыми

я общался, вполне довольны базовым Ubuntu. Делать всё в Emacs по-прежнему популярно, но наблюдается значительный интерес к современным редакторам кода, таким как Atom, или даже — содрогнитесь — Microsoft Visual Studio:Code.

Благодаря контейнерам (будь то chroots, systemd-nspawn, Docker или полноценные VM) для одиночек нет никакой необходимости в отдельном компьютере чисто для разработки. Но крупные совместные проекты используют постоянную интеграцию, и их реальная инфраструктура сборки должна быть централизованной.

Сейчас всё куда менее однородно, чем было когда-то, и открытый код — небольшая часть всего этого. Вы спокойно можете обращаться к серверу Git независимо от своей ОС, рабочего стола или текстовых редакторов. Да даже Microsoft внутри использует его (см. **LXF231**), и (благодаря Windows Subsystem for Linux) их клиентам больше не приходится сражаться с Cygwin, чтобы получить рабочую настройку GCC.

Сайт <https://linuxrig.com>, хотя и не посвященный исключительно разработчикам, предлагает перечни разнообразных систем Linux, что может навести Вас на какие-то мысли.

ПОЛУЧИТЕ ПОМОЩЬ!

Мы рады ответить на все вопросы по Linux, которые вы присылаете на linux.answers@futurenet.com, независимо от их уровня.

У нас у всех когда-то были проблемы, так что не стесняйтесь. Однако мы всего лишь люди (хотя многие подозревают, что Джонни — робот), и очень важно, чтобы вы указывали максимум информации. Если что-то работает на одном дистрибутиве, а на другом — нет, сообщите нам. Если вы получили сообщение об ошибке, пожалуйста, сообщите нам его точный текст и из-за чего оно появилось.

Если у вас есть, или вам кажется, что есть проблема с оборудованием, расскажите нам о нем. Подумайте об установке *Hardinfo* или *Ishw*. Эти программы показывают оборудование на вашем компьютере. Если вы не хотите или не можете установить эти программы, запустите следующие команды в терминале `root` и пришлите нам также файл `system.txt`.

```
uname -a > system.txt
```

```
lspci >> system.txt
```

```
lspci -vv >> system.txt
```

HotPicks

Falkon » Ternimal » Ddrescue » WINE » Polo » Bleachbit
 » Gnome Layout Manager » Daemon-vs-demon » Stupax
 » Fstransform » VLC



Александр Толстой

взлетая на крыльях надежды и желания, помогает воссиять лучшим приложениям с открытым кодом!

WEB-БРАУЗЕР

Falkon

Версия: 3.0

Сайт: <https://github.com/KDE/falkon>

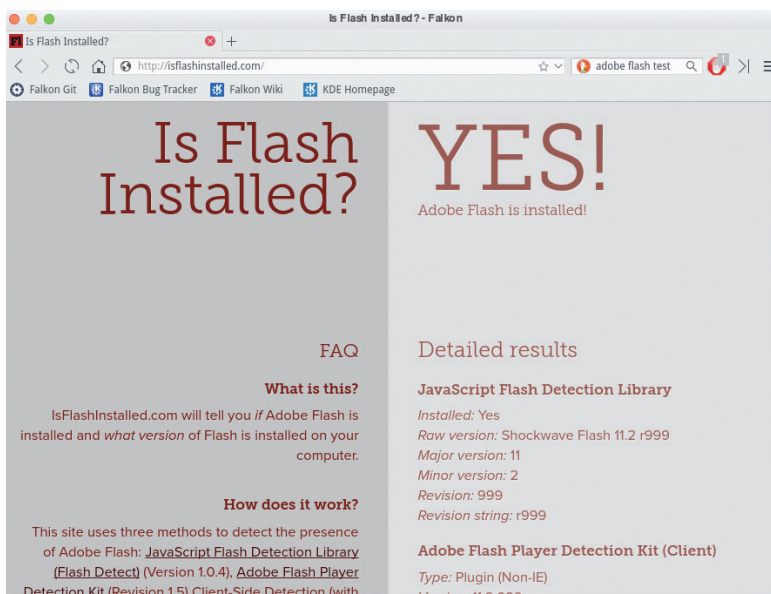
Мы любим альтернативные web-браузеры и браузеры с открытым кодом, особенно когда они способны заменить *Chromium* или *Firefox* для большинства повседневных задач. Богатство выбора особенно важно в современном мире, где, похоже, владеют вышеупомянутые два.

В **LXF212** и **LXF229** мы рассказывали о *Qupzilla*, быстром и богатом функциями web-браузере на QtWebEngine. Будучи реалистами, на сей раз мы не можем игнорировать мир клонов *Chromium*, поскольку QtWebEngine — это слегка измененная и заново укомплектованная пакетами версия движка с открытым кодом *Chromium* (настроенного так, чтобы им могли пользоваться программы *Qt*). Название изменилось из-за неких закулисных действий, в результате которых *Qupzilla* попала под эгиду проекта KDE и превратилась в *Falkon*.

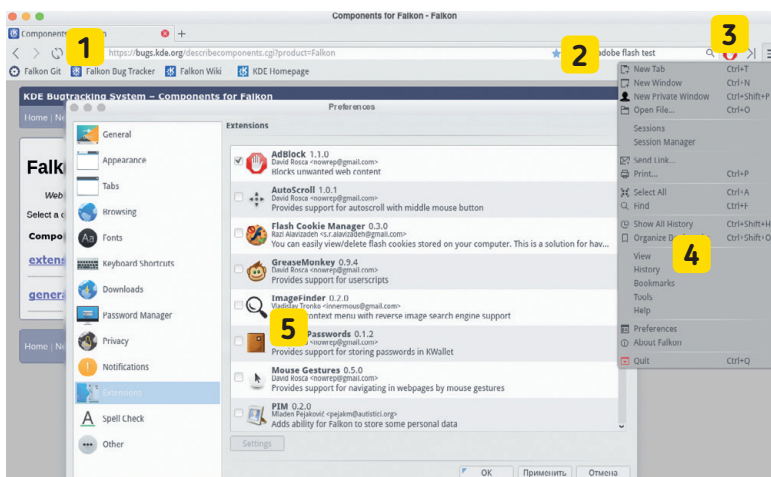
Свежевыпущенный браузер до некоторой степени независим от рабочего стола: он уважает ваши значки и темы и выглядит родным в Windows, Gnome, Plasma и т.д. *Falkon* соединяет производительность *Chromium* и совместимость с (довольно) легким весом и рядом удобных функций по умолчанию — например, встроенным *Adblock*, страницей *Speedial* и инструментом импорта закладок с поддержкой *Firefox* и *Chromium*.

Представители KDE не заявляли об этом во всеуслышание, но естественно будет предположить, что *Falkon* является заменой не работающему браузеру *Konqueror*. Мы использовали *Falkon* неделю-две, и сочли его быстрым и при этом не потребляющим много ресурсов. Он также неплохо справлялся с обработкой разных задач мультимедиа, хотя это скорее зависело от QtWebEngine, чем от самого браузера. Например, некоторые потоковые музыкальные сервисы требуют проприетарных медиа-кодеков, и нам пришлось перекомпилировать QtWebEngine с дополнительными флагами, чтобы включить это в сборку. Учтите, что некоторые дистрибутивы Linux делают это по умолчанию.

В целом, *Falkon* отлично справился с работой и показал себя современным браузером с полезными дополнениями, такими, как поиск зеркального изображения, поддержка индивидуальных скриптов пользователя, персональный информационный менеджер и т.п. Весь список оглашен в Preferences > Extensions [Предпочтения > Расширения].



Знакомимся с интерфейсом Falkon...



- 1 Вкладки и панели**
Falkon предлагает удобное и проверенное временем управление в интерфейсе со вкладками.
- 2 Отдельное поле поиска**
Редкий сейчас элемент – выбор поисковой машины прямо под рукой.
- 3 Блокировка рекламы по умолчанию**
Не надо устанавливать расширение блокировки рекламы – оно уже здесь!
- 4 Кнопки-блины**
Falkon имеет массу функций, доступных по одному добавочному щелчку мыши.
- 5 Готовые расширения**
Часто используемые расширения браузера уже здесь, так что в большинстве случаев настраивать *Falkon* вам не придется.

ИМИТАЦИЯ ФОРМ ЖИЗНИ

Ternimal

Версия: 0.1.0

Сайт: <https://github.com/p-e-w/ternimal>

Нам всегда нравилось пробовать приложения, не имеющие практической пользы, но зато позволяющие интересно провести час-другой. Наглядный пример — *Ternimal*.

Обычно через пару минут ваши глаза обнаруживают в этом имени опечатку, но на самом деле ее там нет. *Ternimal* — это гибрид эмулятора терминала [terminal] и животного [animal], и вы быстро убедитесь, что название хорошее. Программа рисует анимированного червя или змею прямо в окне терминала. Это существо создается из прямоугольных символов Unicode, и хотя сама картинка неизбежно пикселист, анимация очень естественная и плавная.

Практической пользы у *Ternimal* никакой — разве что удивить друзей; но важно отметить, что это приложение считается упражнением в программировании на Rust. *Ternimal* был написан на Rust, а поскольку это очень компактное приложение, вы можете получить бесценный опыт, изучая его код.

Запустить программу очень легко. Надо всего лишь установить общий пакет 'rust' в своем дистрибутиве Linux. Rust



Поиграв с настройками, мы в конечном итоге поместили команду в .bashrc.

поставляется с компилятором *rustc*, для которого следует скомпилировать исполняемый файл:

```
$ rustc -O ternimal.rs
```

При запуске без аргументов *Ternimal* рисует жирного бледного головастика, который вслепую движется в черной среде (ну или какого цвета ваш фон терминала). Приложение не предлагает встроенной помощи, но его разработчик любезно показал несколько вариантов, которые имитируют анаконду, стаю мелких рыбок или даже радугу. Попробуйте, например, это:

```
$ ternimal length=200 thickness=0,4,19,0,0
```

Вы также можете присоединить другие аргументы, например, **radius** [радиус], **gradient** [градиент] и **padding** [заполнение]. Изучите эти опции и другие, и весело проводите время с *Ternimal*.

ИНСТРУМЕНТ ВОССТАНОВЛЕНИЯ ДАННЫХ

DDrescue

Версия: 1.23

Сайт: www.gnu.org/software/ddrescue

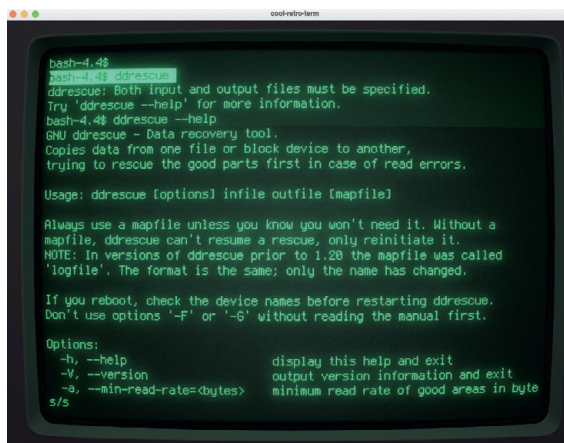
Linux — отличная ОС для восстановления данных. А *DDrescue* — известная и уважаемая программа восстановления; также это классическое приложение в стиле UNIX, которое поддерживается проектом GNU. Новые версии программы появляются практически раз в год.

Цель *DDrescue* — спасти данные с поврежденных жестких дисков и оптических носителей. Обратите внимание, что приложение работает только с механическими HDD-дисками и CD/DVD; SSD или флешки остались за бортом.

DDrescue может восстановить информацию с дисков с поврежденными секторами и иных средств хранения, которые не удается смонтировать. Впервые встретив 'read error [ошибку чтения]', ядро Linux часто отказывается читать таблицу разделов, и вы ничего не видите. В такой ситуации *DDrescue* пригодится, чтобы извлечь данные с устройства в файл ISO. Пример команды:

```
# ddrescue -d -r3 /dev/sda test.img test.mapfile
```

Опция **-d** означает прямой доступ к диску (игнорируя кэш ядра), а **-r3** означает три попытки для каждого обнаруженного поврежденного сектора. Возможно, в первый раз вы захотите пропустить опцию **-f** при запуске команды над поврежденным диском, поскольку повторяющиеся попытки доступа к поврежденным секторам могут привести к полному отказу устройства до того, как вам удастся что-то из него извлечь.



Не выбрасывайте поврежденные диски, пока не запустите над ними *ddrescue*.

mapfile (он же — **logfile**, в старых версиях *DDrescue*) — важный компонент для слияния нескольких IMG-файлов в один. Это отличная функция, позволяющая комбинировать результаты нескольких попыток спасения вашего диска. Для этого используйте следующее:

```
# ddrescue -m test.mapfile test1.img test2.img merge.mapfile
```

Если ваш диск периодически демонстрирует признаки жизни, есть неплохой шанс, что вы получите кумулятивный образ любых данных, оставшихся на нем. То же относится к поцарапаным дискам.

ЭМУЛЯТОР ПРОГРАММ

WINE

Версия: 3.3

Сайт: www.winehq.org

История запуска бинарников Windows в Linux постоянно развивается с момента появления *WINE* в 1993 г. Изначально предполагалось, что людям надо запускать рабочие и производственные программы, которых пока что не было в Linux, и поэтому *WINE* стал известен как инструмент для запуска *Office* или *Photoshop* вне Windows.

Недавно реанимированный дистрибутив Linspire раньше назывался Lindows и включал *WINE* как убойную и готовую к работе функцию, хотя позднее разработчики от нее отказались. Современные версии *WINE* справляются с большинством основных программ Windows очень хорошо, и следующая очевидная цель — привести высококачественные игры Windows в Linux. В самом деле, недавние релизы *WINE*, особенно серии 3.x, вызвали большой интерес среди геймеров.

Решив посмотреть новинки версий 3.1–3.3, вы увидите, что закрыто множество проблем, которые касаются современных популярных игр, таких как распространяемые через Valve Steam и пока не портированные в Linux. *WINE* 3.3 предлагает улучшенную поддержку геймпада благодаря его драйверу HID и начальную поддержку Vulkan, а также новых многовыборочных [multisample] текстур для Direct3D. И не пропустите самый последний код *WINE* от проекта *Wine Staging* (<http://github.com/wine-staging>), чтобы воспользоваться самыми свежими улучшениями



Вам не нужна двойная загрузка с Windows, чтобы запускать программы win32 или win64.

производительности графики и такими дополнительными прелестями, как поддержка тем *GTK3*. Главное ПО *WINE* наконец-то отказалось от старых частей DOS. Это означает, что если вы планируете запускать доисторические программы DOS в Linux, вам лучше использовать *Dosbox*.

Большинство файлов *.exe* запускаются автоматически, если они связаны с *WINE*, но если вы щелкаете по приложению — и ничего не происходит, обязательно запустите `$ wine app.exe` из командной строки и проверьте результат. Как обычно, команда `$ winecfg` является дверью в волшебный мир настроек *WINE*, где можно изменить множество настроек производительности и совместимости.

МЕНЕДЖЕР ФАЙЛОВ

Polo

Версия: 18.2 Beta

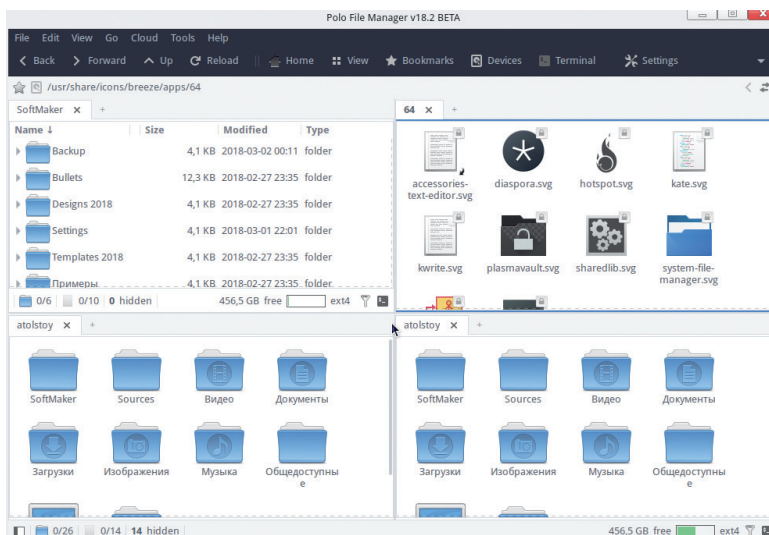
Сайт: <https://github.com/teejee2008/polo>

Если вас утомили менеджеры файлов с классическими значками или с двумя панелями, обратите внимание на *Polo*. Это современный менеджер файлов для Linux, написанный на Vala и основанный на *GTK3*. У него есть ряд отличительных свойств, которые определенно стоит исследовать.

Мастер стилей предлагает вам несколько удобных видов. Вы можете использовать *Polo* с «экстремальным» стилем, который даст вам четыре панели в сочетании с отображением в виде списков, общим окном пути и общей строкой состояния. Доступен также двухпанельный режим командера.

Мы немного поиграли с *Polo* и нашли, что он интуитивен в использовании. Например, в нем имеются вкладки, диаграммы свободного места на монтированных дисках, поддержка облачного хранения (для этой функции требуется *Rclone*), менеджер сессии, встроенная панель терминала и множество других деталей, с которыми вы рано или поздно столкнетесь, если решите установить программу.

Внутри контекстного меню находятся быстрые ссылки для монтирования файлов ISO, запись на USB-брелки, изменение PDF и графических файлов изображений и даже управление экземплярами виртуальных машин KVM.



Polo создает впечатление универсального инструмента, хотя проект быстро развивается и мы ожидаем, что в ближайшем будущем он еще пополнится. У приложения очень ясная структура, и после некоторого времени использования *Polo* возврат к *Nautilus* был как пересадка со скоростного поезда в вагон, прицепленный к паровозу. Мы считаем, что *Polo* легко может стать вашим основным менеджером файлов, особенно в среде рабочего стола на *GTK3*.

В проекте имеются прекомпилированные Deb-пакеты для семейства Ubuntu, а также универсальные установщики *.run*, которые поладят с любым дистрибутивом Linux.

Четырехпанельный режим похож на мультиплексор, примененный к управлению файлами.

ПРОГРАММА ОЧИСТКИ СИСТЕМЫ

Bleachbit

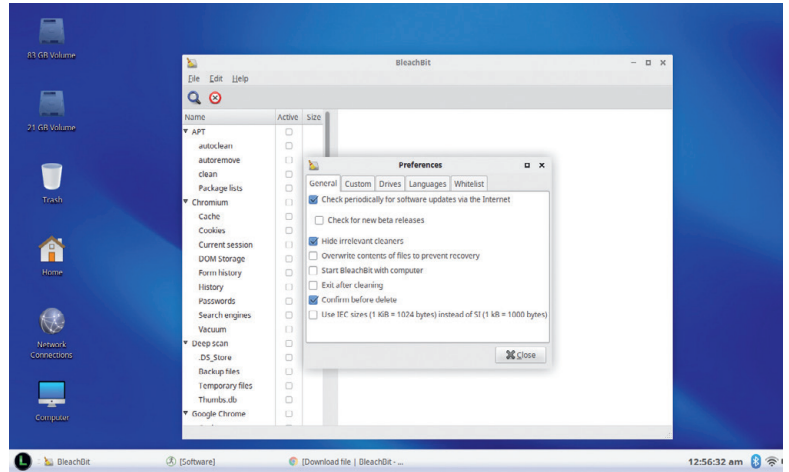
Версия: 2.0

Сайт: <https://github.com/bleachbit/bleachbit>

Многие люди постоянно используют один дистрибутив Linux, многие — периодически всё меняют, но каждый раз используют одну и ту же директорию `/home` на отдельном разделе. Подобные пользователи, вероятно, знают, каково это — иметь кучу хлама и временных файлов в своем профиле. Хотя вы всегда можете проверить и удалить ненужные файлы вручную, есть простой инструмент GUI, который за этот свой побочный эффект получил название *Bleachbit* [англ. «отбеливатель битов»].

Суть этой программы похожа на то, что *CCleaner* делает для Windows. *Bleachbit* — дружелюбно выглядящий инструмент с простым интерфейсом, и, практически, защитой от дурака. Т.е. он сообщает вам о возможных последствиях, когда вы активируете определенные рычаги и триггеры в левой стороне окна.

Обратите внимание, что *Bleachbit* не ограничивается очисткой только `/home`. Он также может почистить кэш пакетов *Apt* для всей системы, выцарапав немного добавочного свободного места в разделе `root`. Новая версия релиза 2.0 предлагает солидную улучшенную поддержку очистки профилей *Chrome* и *Chromium* (которые склонны за недели активного использования накапливать огромное количество данных) и новую функцию безопасного



удаления баз данных *SQLite*. Как и прежде, *Bleachbit* является идеальным решением для очистки кэша *Firefox* и *Thunderbird*, удаления неиспользованных пакетов локализации и таких файлов, как *Thumbs.db* и *.DS_Store*.

С помощью *Bleachbit* также можно безопасно удалять файлы: скомандуйте ему удалить файлы и переписать свободное место на диске пустыми данными за один проход, обеспечив невозможность восстановления данных с помощью известных инструментов типа *Photorec* или *Testdisk*. Разработчики *Bleachbit* настаивают, что одного прохода будет достаточно, а последующие операции по перезаписыванию разве что укрепят ваш душевный покой, но не более того. Так или иначе, *Bleachbit* должен быть первым выбором для тех пользователей, которым надо безопасно удалить важные данные без возможности восстановления.

Bleachbit может очистить и удалить безвозвратно, да добавок сам себя обновляет!

ИНСТРУМЕНТ НАСТРОЙКИ

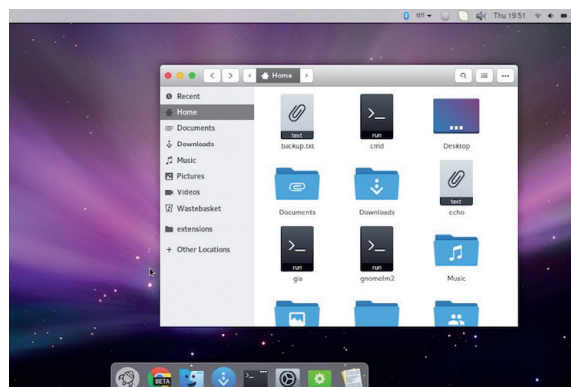
Gnome Layout Manager

Версия: Git Сайт: <https://github.com/vmavromatis/gnome-layout-manager>

Разногласия между поклонниками разных рабочих столов никогда не закончатся, но мы здесь во имя мирного диалога. Фанаты рабочего стола *Gnome Shell* от *Red Hat* определенно полюбят небольшой, но очень полезный скрипт, который мы выбрали для обзора. *Gnome* является флагманским рабочим столом как минимум для двух основных дистрибутивов Linux — *Fedora* и *Ubuntu* — и скрипт *Gnome Layout Manager* определенно привлечет интерес тех, кто хочет точно настроить свою систему для наилучшего внешнего вида и работы.

Gnome Layout Manager автоматизирует многие обычные задачи, которые люди выполняют, чтобы их рабочий стол выглядел, как *Unity*, *Windows* или *macOS*. Вам больше не надо устанавливать бесчисленные расширения *Gnome Shell* и погружаться в глубины *Dconf-editor*, чтобы настроить всё по своему вкусу. *Gnome Layout Manager* автоматизирует этот процесс и проведет вас по списку вопросов, где вам надо будет просто указать, как вы хотите, чтобы выглядел ваш рабочий стол.

Изменения касаются размещения дока приложений, панели инструментов, тем *GTK3* и *Gnome Shell*, а также соответствующего набора дополнений с <https://extensions.gnome.org> для каждого вида. Если вы используете новый *Ubuntu 17.10* или *18.04*, который



Благодаря этому отличному скрипту можно запросто имитировать другую операционную систему.

по умолчанию предлагает индивидуальную настройку *Gnome*, вам не надо ничего делать, чтобы «перекрасить» его под *Unity*, однако со внешним видом *Windows 10* и *macOS* определенно стоит повозиться.

Поскольку это не программа, которую надо компилировать из исходника или искать версию в пакетах, всё, что вам надо сделать — отметить скрипт как исполняемый и запустить его:

```
$ chmod +x layoutmanager.sh && ./layoutmanager.sh
```

Всё остальное довольно просто и даже включает уведомления рабочего стола (через *Zenity*).

ИГРА ТИПА РВИ-И-МЕЧИ

Daemon-vs-Demon

Версия: Git Сайт: <https://github.com/securas/Daemon-vs-Demon>

В этой игре вам придется сражаться с демонами — злобными духами, которые норовят убить вашего героя. Вы играете за демона, который тоже сверхъестественный дух, но не злобный, а как раз добрый до мозга костей. В *Daemon-vs-Demon* злая судьба привела нашего героя в глубины ада, и теперь ему приходится убивать гадких демонов, чтобы вернуться в мир живых.

Игровой мир унылый и намокший от дождя. Демоны могут нападать из тени, заставляя вас врасплох и быстро нанося вам раны, без шанса среагировать. Интересно, что ваш демон может принимать форму побежденных демонов и оставаться незамеченным, путешествуя по карте. Каждая трансформация длится десятки секунд, и как только вы становитесь собой, стаи демонов набрасываются на вас со всех сторон. Цель *Daemon-vs-Demon* — найти выход из каждого уровня, что может быть нелегкой задачей даже без присутствия этих назойливых демонов. Игра основана на движке Godot, и если вы помните чудесную игру *Mouse Boat* из **LXF219/220**, вы заметите между ними некоторое сходство.

Daemon-vs-Demon предлагает грубую пиксельную графику — и играется в эту игру, как в ретро-платформер. Она победила



Подойдите к этим окровавленным останкам и нажмите К, чтобы ненадолго превратиться в демона.

на Game Off 2017, ежегодном игровом джеме от Github в честь открытого кода, так что мы настоятельно рекомендуем познакомиться с этой игрой, поскольку она предлагает захватывающую интригу в отличном стиле. Вам ничего не надо компилировать — просто скачайте версию Linux с <https://securas.itch.io/daemon-vs-demon> и запустите единственный исполняемый файл внутри.

ИГРА С БОКОВОЙ ПРОКРУТКОЙ

Stupax

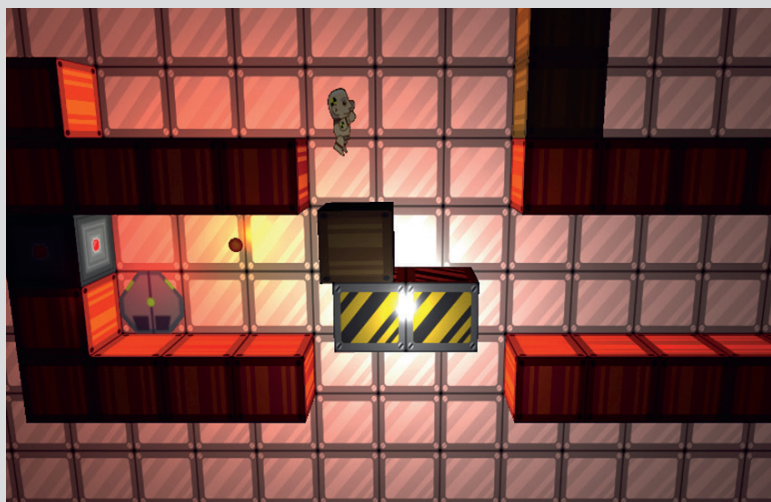
Версия: 3.0.1

Сайт: <https://github.com/mbarde/stupax>

Большинство современных игр на HTML5 — с закрытым кодом, однако лежащие в их основе движки и библиотеки свободны. Такие игры часто разрабатываются ради коммерческого успеха в Google Play или Appstore, и трудно найти проект с полностью открытым кодом, который бы стоил вашего времени. *Stupax* — один из таких проектов: на самом деле, нас поразило его качество.

Мы бы описали *Stupax* как аркаду с горизонтальной прокруткой. Вы управляете подвижной платформой, чтобы помочь персонажу пройти 23 уровня с постоянно увеличивающейся сложностью навигации. В некотором смысле *Stupax* переворачивает всё с ног на голову, потому что вы управляете уровнем, а не героем (который движется автоматически). Ему нужно добраться до красной двери, и ваша задача — не дать ему упасть в пропасть или умереть, будучи пробитым пушечным ядром.

Герой чрезвычайно глуп! Он не способен прыгать, нагибаться или отстреливаться, так что вам надо очень плотно его опекать. Он просто-напросто без затей помнется вперед и меняет направление, только столкнувшись с препятствием. Платформа, которой вы управляете, движется быстрее героя, так что у вас будет достаточно времени, чтобы сменить позицию и поймать его, если он упадет.



Первые уровни *Stupax* довольно просты, но потом вы сталкиваетесь с задачами потруднее, где вам придется решать некие головоломки и находить путь в лабиринте. Чем дальше вы продвигаетесь в игре, тем больше она становится гибридом *Sokoban* и классического платформера. Графика, звук и геймплей отличные, и делают игру очень увлекательной.

В *Stupax* можно играть прямо на <http://mbarde.bplaced.de/stupax>, но поскольку у нее открытый код, ничто не мешает вам развернуть собственный игровой сервер (локальные сервера *Apache* или *Nginx* отлично подойдут). Игра написана на JavaScript (BabylonJS, если быть точными) и предлагает два файла **.html**: один для игры, а другой для редактора уровней.

Всё движется, и вам надо отслеживать несколько объектов одновременно.

ИНСТРУМЕНТ ФАЙЛОВОЙ СИСТЕМЫ

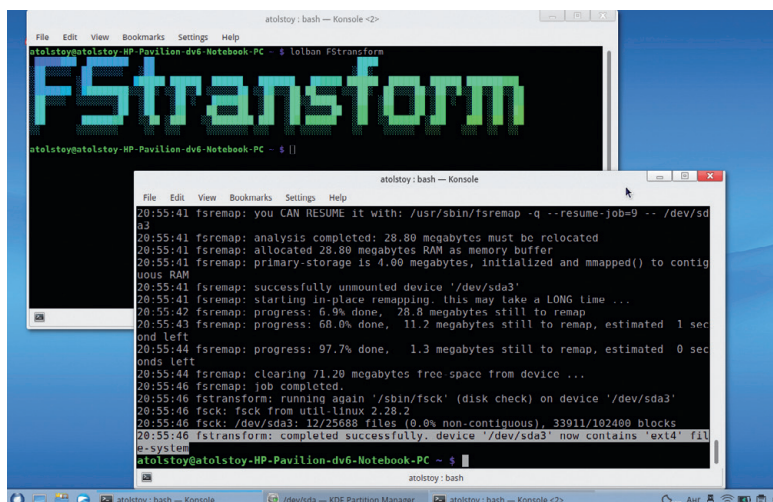
Fstransform

Версия: Git Сайт: <https://github.com/cosmos72/fstransform>

Инструмент Linux, который колдовал с файловой системой, мы в последний раз видели более десяти лет назад, и это был, вероятно, скрипт *Bash* от Кона Коливаса [Kon Kolivas] для дефрагментации. Он помогал перераспределять биты данных на жестком диске.

Fstransform — не менее потрясающий, ведь эта программа конвертирует файловые системы из одной в другую, поддерживая ext2, ext3, ext4, reiserfs, jfs и xfs в любых комбинациях. Вы можете трансформировать свой раздел в другой формат прямо на месте, не теряя данных. Как и полагается, автор многократно вас предупреждает, что вы всё делаете на свой страх и риск и в некоторых случаях можете безвозвратно потерять свои данные, и что программа полностью экспериментальная и ничего вам не гарантирует...

Мы провели собственные тесты с небольшим учебным разделом и обнаружили, что *Fstransform* работает блестяще, если вы тщательно соблюдаете его ограничения. У вас должно быть достаточно свободного места, да проверьте, что исходная ФС поддерживает разреженные файлы (или просто не помещайте в него такие файлы). Настоятельно рекомендуем прочитать



довольно обширный **Readme.md** (он того стоит), чтобы выяснить подробности, прежде чем запускать нечто вроде

```
$ sudo fstransform /dev/sda3 reiserfs
```

Как обычно, замените имя раздела и тип файловой системы на то, что вам нужно. Кстати, *Fstransform* поддерживает и другие, менее популярные файловые системы, для которых вам надо будет добавить опцию в команду:

```
$ sudo fstransform --force-untested-file-systems /dev/sda3 ntfs
```

Если вы планируете конвертировать реальные данные с помощью *Fstransform*, запустите скрипт Кона Коливаса (его можно найти на <http://bit.ly/2F6E3wv>) во избежание повреждения данных при переходе с одной файловой системы на другую.

Если вы готовы рискнуть, то конвертирование NTFS в Ext4 — быстрый процесс.

МЕДИА-ПЛЕЙЕР

VLC

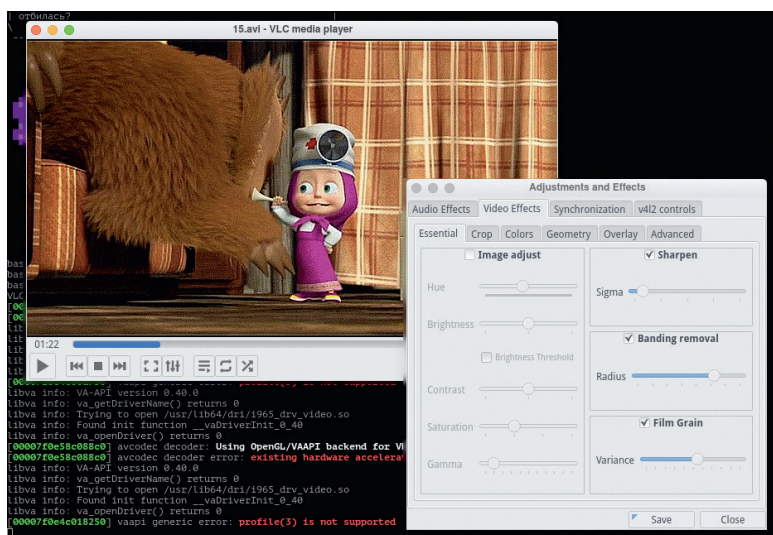
Версия: 3.0.1

Сайт: <https://www.videolan.org>

VLC — важная программа для каждой установки Linux на тот случай, если вы захотите послушать музыку или посмотреть видео. Причем это универсальное решение для множества вспомогательных задач, помимо воспроизведения. *VLC* умеет перекодировать файлы из одного формата в другой, перепакетывать медиа-контент с определенным кодеком, воспроизводить поток и усиливать звук аудио свыше 100% с помощью знакомой панели управления, окрашенной в цвета радуги — что позволяет исправлять слишком тихие аудиотреки.

Слухи о третьем основном релизе *VLC* ходили с ноября прошлого года, и это очень радовало многих, поскольку после релиза 2.2.0 в 2015 г. новостей о *VLC* не было. *VLC 3.0* впечатлил нас значительно улучшенным видео VA-API, которое позволяет смотреть фильмы 4K со скромной нагрузкой на CPU в размере 20%. *VLC* использует OpenGL для рендеринга движущейся картинки и работает очень плавно даже на маломощных системах.

VLC 3.0 может предложить еще больше ценителям звука Hi-Fi и мультимедиа благодаря поддержке кодеков E-AC3, TrueHD и DTS-HD и нового панорамного изображения (используйте клавиши со стрелками для управления камерой в 360-градусной панораме) плюс множеству переписанных медиа-декодеров. Вы увидите, как быстро *VLC 3.0* теперь осуществляет



поиск по большим видео, и может получать доступ к большому числу локаций для заполнения плей-листа, в том числе к вашему устройству Android через протокол MTP. Заметьте, что *VLC for Android* тоже был обновлен.

Как и раньше, *VLC* поставляется с собственным набором кодеков, так что он не слишком зависит от таких вещей, как установка *GStreamer* на уровне всей системы. После установки *VLC* вам больше не придется беспокоиться о поддержке популярного медиа. И это хорошо. **LXF**

Не забудьте, что можно улучшить качество изображения с помощью встроенных видеозащитных эффектов.

На диске

Дистрибутивы, приложения, игры, книги и всякое-разное...



НЕЙЛ БОТВИК

В этом месяце я сражался с *Grub*. В старые добрые времена файла настройки из пяти строк было вполне достаточно, чтобы загрузить дистрибутив Linux. А теперь у нас массивный *Grub2* с файлами настройки более чем на 100 строк, набитый подгружаемыми модулями и операторами условий. *Grub2* больше подходит для автоматического создания своих меню, благодаря чему идеален для программ установки дистрибутивов. Однако и создавать меню вручную никогда еще не были настолько просто.

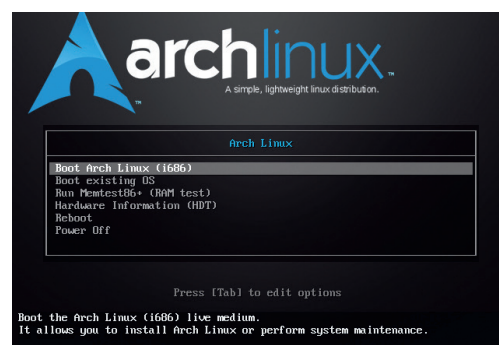
RTFM, я слышу твой зов! Да, документация довольно обширная, но иногда ей не хватает информативности. Это классическая ситуация с программой, которая становится мощнее, но при этом и сложнее, тогда как документация не успевает за ней. Мне удается чувствовать и любовь и ненависть одновременно — а не что-то одно — особенно когда я обнаруживаю интересные функции, о которых вам никто не рассказывает.

Если вы уже загружали DVD этого месяца, вы увидите разную графику у подменю каждого дистрибутива. Это было более или менее случайным открытием, когда я попытался всего лишь сменить текст подсказки в одном меню. Решение оказалось сложнее, чем можно было ожидать, зато открыло путь к симпатичной новой функции на диске LXF.

ARCH ДЛЯ СТАРЕЙШИН

Arch Linux 32-2018.02

Число дистрибутивов, поддерживающих 32-битные ПК, продолжает сокращаться. Теперь уже и Arch Linux отказался от поддержки устаревшего оборудования. Однако сообщество взяло бразды правления в свои руки и теперь поддерживает 32-битные (i686) пакеты и программу установки. Arch32 — стандартный установщик Arch Linux. Иными словами, вам придется выполнять работу самим, зато в конечном итоге вы получите дистрибутив, который хотите, а не тот, который вам нужен по мнению некоего разработчика. В этом суть Arch Linux: вы сами делаете выбор, однако реализовывать этот выбор вам тоже приходится самостоятельно. Руководство по установке см. в wiki.archlinux.org/index.php/installation_guide.



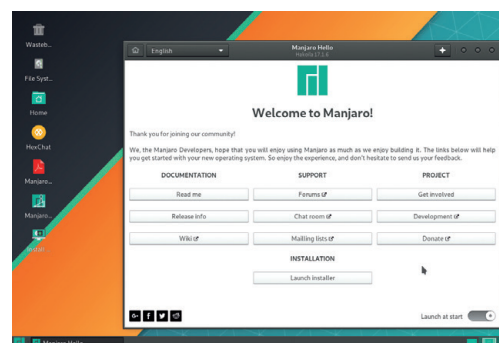
Слухи о его смерти сильно преувеличены — Arch Linux для 32-битных CPU жив-здоров.

КРАСИВОЕ ЛИЦО ARCH

Manjaro Xfce-17.1.6

Простота настройки Arch делает его хорошим выбором в качестве системы сборки для других дистрибутивов, о чем свидетельствует рост числа дистрибутивов, которые сейчас приняли Arch Linux за основу. Manjaro — один из наиболее популярных таких дистрибутивов. Имеется несколько настольных версий Manjaro, и наша — с Xfce. Если не считать изначального рабочего стола, версии одинаковы, и при желании вы можете установить другой рабочий стол.

Manjaro поставляется с простым в использовании установщиком, который устанавливает и настраивает всё, чтобы вы получили готовый графический рабочий стол, благодаря чему Manjaro намного проще установить, чем Arch. Одно из ключевых свойств, которое он унаследовал от своего предка — это возобновляемый релиз. Вместо выпуска совершенно нового дистрибутива каждые шесть–девять месяцев, Manjaro регулярно обновляется по мере появления новых версий пакетов. А значит, вы получаете новые программы



Manjaro — это Arch Linux, но проще и доступнее, с разными рабочими столами. Здесь — версия Xfce.

быстрее, чем в дистрибутивах, практикующих более типичную модель релизов. Более того, у вас нет основного обновления, которым надо озаботиться на очередной стадии цикла релизов.

ВНИМАНИЕ! ЭТО ВАЖНО!

ПОРЧЕННЫЕ ДИСКИ: В маловероятном случае какого-либо дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу disks@linuxformat.ru или телефону +7 (812) 309-0686.

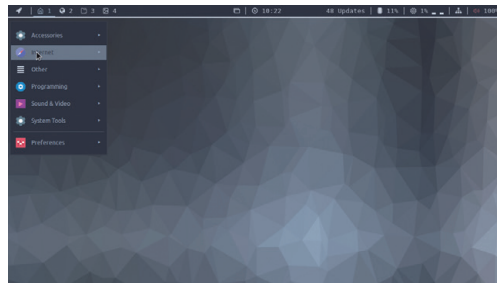
ЛЕГКОВЕСНЫЙ ARCH

ArchLabs-2018.02



Ил да был когда-то легковесный быстрый дистрибутив по имени CrunchBang. Увы, разработка его прекратилась несколько лет назад, но потом вновь началась в виде BunsenLabs. Оба были основаны на Debian. Всё было хорошо, пока группа разработчиков не решила, что им нравится концепция и вид CrunchBang/BunsenLabs, но заодно хочется еще гибкости базы Arch Linux; так появился на свет ArchLabs.

ArchLabs использует менеджер окон *OpenBox* и ясную, минимальную тему, чтобы обеспечить вам легковесный, быстрый и при этом полнофункциональный рабочий стол. Чтобы установить ArchLabs, щелкните правой кнопкой по рабочему столу и выберите опцию Install.



Минимальный и быстрый, но при этом умелый – ArchLabs отказался от внешнего лоска в пользу производительности.

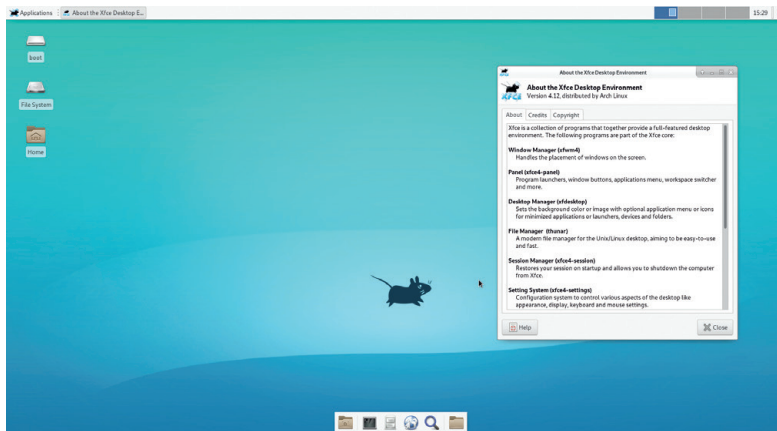
ARCH ДЛЯ Pi

Arch Linux для Raspberry Pi 2 & 3

С толь легковесный дистрибутив, как Arch Linux, является очевидным кандидатом на всемирно любимый компьютер размером с кредитку. Поэтому неудивительно, что у нас имеется версия Arch для Raspberry Pi. Это часть проекта Arch Linux ARM, портированного Arch на разные платформы для ARM.

Если вы уже посмотрели на DVD, то увидите, что процесс установки не так прост, как для Raspbian, хотя по крайней мере устраняет риск копирования образа не на то устройство с помощью dd. Установка Arch на Pi

включает разбиение на разделы и форматирование SD-карты (с помощью вашего компьютера, а не Pi), извлечение архива на нее и перемещение новых файлов. Инструкции — на DVD и сайте проекта. Затем вы можете вставить карту в свой Pi и загрузиться с нее. У вас будет только командная строка: это традиция. Вам придется установить X-сервер и рабочий стол, если они вам вообще нужны. Информацию о проблемах, связанных именно с Pi, вы найдете на archlinuxarm.org/platforms/armv6/raspberry-pi.



Arch Linux, запущенный на Raspberry Pi, здесь – с рабочим столом Xfce, идеально подходящим к возможностям оборудования Pi.

» НОВИЧОК В LINUX? НАЧНИТЕ ОТСЮДА!

Мы ответим на ваши вопросы, установите DVD и откройте [index.html](#).

- Что такое Linux? Как его установить?
- Есть ли эквивалент MS Office?
- Для чего нужна командная строка?

СКАЧАЙТЕ
СВОИ ОБРАЗЫ
DVD!

Найдите код и образы DVD на www.linuxformat.com/archives

» И ЕЩЕ!

PROGRAMMING FROM THE GROUND UP — ПРОГРАММИРОВАНИЕ С САМЫХ ОСНОВ

Это введение в программирование, но оно отличается от большинства вводных статей в одном важном отношении: выборе языка. Большинство введений в программирование используют C по причине его повсеместного распространения, или язык, который прост для понимания и отладки и прощает ошибки новичков — типа Python.

Эта книга использует язык, который не относится ни к одной из групп: ассемблер. Хотя это может показаться странным выбором для введения в программирование, ассемблер предлагает прямой доступ к системе, и все выученные вами здесь уроки останутся с вами, если в будущем вы решите использовать компилируемый или интерпретируемый язык программирования.

КНИЖНАЯ ПОЛКА

- **Расширенное руководство по скриптам Bash** Изучите написание скриптов еще лучше.
- **Руководство по Bash для начинающих** Осваивайте написание скриптов *Bash*.
- **Руководство по созданию скриптов Bourne Shell** Начинайте 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.
- **Нырните в Python** Всё, что вам надо знать.
- **Словарь Linux** Linux от А до Я.
- **Linux Kernel in a Nutshell [Ядро Linux в двух словах]** Введение в ядро, написанное великим магистром ядра Греггом Кроа-Хартманом [Greg Kroah-Hartman].
- **Руководство системного администратора Linux** Контролируйте свою систему.
- **Обзор инструментов** Полный обзор инструментов GNU.

ПРОШЛЫЕ НОМЕРА » ПРОПУСТИЛИ?

LXF235 Апрель 2018



- Отгоним шпионов Все виды самозащиты
- Как стать телеведущим Видеоконференции
- Linux — фотограф Шедевры и их хранение
- Открываем лавочку Интернет-магазин на WordPress

LXFDVD

Anarchy, LXLE, Parrot Security OS, Solus, Tails, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_235/
PDF-версия: shop.linuxformat.ru/elxf_235/

LXF234 Март 2018



- Экспресс-курс Linux 60 минут — и вы в дамках
- Не только навигация Выберем менеджер файлов
- Терминал под музыку Работать будет весело
- В порту по случаю Нарастиваем энтропию

LXFDVD

AV Linux, Bodhi, elementary OS, Kali Light, ROSA, 11 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_234/
PDF-версия: shop.linuxformat.ru/elxf_234/

LXF233 Февраль 2018



- ПК своими руками От железа до софта
- Android на Linux Выбираем эмуляторы, пишем приложения
- Едет, едет почта Почтовые клиенты живы
- QR-код с пингвином Или с другой картинкой

LXFDVD

Fedora, Linux Mint, Netrunner, OpenMediaVault, Tails, 11 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_233/
PDF-версия: shop.linuxformat.ru/elxf_233/

LXF232 Январь 2018



- 32-битные дежавю Лебединая песня?
- Споке и его друзья Дистрибутивы-гномофилы
- Linux сладкозвучный Аудио и еще раз аудио
- Скажи пароль Ни за что не говорите!

LXFDVD

AntiX, Debian, Legacy OS, Linux Lite, Manjaro, Peppermint, Tiny Core, Zorin OS, 11 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_232/
PDF-версия: shop.linuxformat.ru/elxf_232/

LXF231 Декабрь 2017



- Заметки анималиста Творческий Трубказуб!
- Шеренга столов Работай — не хочу
- Доктор вернулся В формате импортозамещения
- Точки доступа Обезопасим с OpenVPN

LXFDVD

Lubuntu, Q4OS, RevengeOS, Robolinux, Ubuntu, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_231/
PDF-версия: shop.linuxformat.ru/elxf_231/

LXF229/230 Ноябрь 2017



- Raspberry Pi как телохранитель Защищает нашу сеть
- Шифруемся Иначе теперь не проживешь
- Немного о сборках Доверяем только воспроизводимым
- Пакеты будущего Независимо от дистрибутивов

LXFDVD

ArchLabs, Backbox, Bodhi Legacy, Sparky, LinHES, OpenELEC PC and RPi, Rescatux, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_229-230/
PDF-версия: shop.linuxformat.ru/elxf_229-230/

Закажите на сайте www.linuxformat.ru

Подпишитесь на печатную версию журнала на www.linuxformat.ru/subscribe или www.linuxcenter.ru, и получите электронную версию в подарок!

Санкт-Петербург: (812) 309-0686 Москва: (499) 271-4954

Linux Format ВКонтакте: vk.com/linuxform



Содержание

Страница 0

Дистрибутивы

Arch 32 2018.02 (32-битный)
Разрабатываемая сообществом версия Arch для старых машин. Предназначена для опытных пользователей, которые сами делают выбор и сами его реализовывают.

ArchLabs 2018.03 (64-битный)
Самообновляемый дистрибутив, потомок Vynsen Labs, перешедший на базу Arch Linux, с менеджером окон Openbox в качестве основного рабочего стола.

Arch Linux ARM (32-битный)
Версия Arch, специально оптимизированная под Raspberry Pi.

Manjaro 17.1 Xfce (64-битный)
Быстрая и удобная настольная ОС на базе Arch Linux с легковесным рабочим столом Xfce. Ключевые слова — интуитивно понятный процесс установки, автоматическое обнаружение оборудования, стабильная возобновляемая версия, возможность установки нескольких ядер, специальные скрипты для управления графическими драйверами и исключительная настраиваемость.

Hotricks

bleachbit 2.0 Программа очистки системы
bleachbit.com/bleachbit/bleachbit/

Daemon-ns-Demol (GN) Игровая титла «Вин-и-мечи»
bleachbit.com/bleachbit/bleachbit/bleachbit/

Driftnet 1.23 Инструмент восстановления данных
www.driftnet.org/software/driftnet/

Falkon 3.0 Web-браузер
bleachbit.com/KDE/falkon/

Fstoolsnm (GN) Инструмент файловой системы
bleachbit.com/bleachbit/bleachbit/bleachbit/

Gnome Layout Manager (GN) Инструмент настройки
bleachbit.com/bleachbit/bleachbit/bleachbit/bleachbit/

Pio 18.2 Beta Менеджер файлов
bleachbit.com/bleachbit/bleachbit/bleachbit/

Slurpx 3.0.1 Игра с боковой прокруткой
bleachbit.com/bleachbit/bleachbit/bleachbit/

Terminal 0.1.0 Имитация форм жизни
bleachbit.com/bleachbit/bleachbit/bleachbit/

VLC 3.0.1 Медиа-плеер
www.videolan.org

Wipe 3.3 Мультиязычная программа
www.wipefs.org

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. И 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 всё же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru



Arch 32
2018.02
Собран для старых ПК
32-БИТНЫЙ



ArchLabs
2018.03
Новый пацан во дворе
64-БИТНЫЙ



Arch Pi
Для Pi 2/3
Оптимизирован под ARM
32-БИТНЫЙ

НОВИНКА!

МАНДЖАРО LINUX 17.1 64-БИТНЫЙ

Дистрибутив на базе Arch для всех уровней

- Самообновляемый релиз
- Мощный и дружелюбный
- Классический легковесный рабочий стол Xfce

Содержание

Учебники

CodeMachine
Python

Помощь

Руководство новичка
Ответы
Rite Знаменитый справочник

Главное

Grub
Mentest86+

Документация: 12 книг о Linux (на английском языке)

Advanced Bash Scripting Guide Подробное руководство по программированию на Bash
Bash Guide for Beginners Руководство по Bash для начинающих
Bourne Shell Scripting Начальное руководство по программированию на Bash

The Cathedral and the Bazaar Классический текст Эрика Реймонда [Eric S Raymond] «Собор и базар»
The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian
Dive Into Python Учебник по программированию на Python
Intro to Linux Начальное руководство по Linux
Linux Dictionary Словарь Linux, объясняющий специфическую терминологию
Linux Kernel in a Nutshell Описание ядра Linux, созданное одним из его выдающихся разработчиков — Грегом Кроа-Хартманом [Greg Kroah-Hartman]
Producing OSS Как выпускается свободное ПО
Programming from the Ground Up Программирование с самых основ (язык ассемблера)
System Administrators Guide Руководство по базовому администрированию Linux
GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

Пожалуйста, перед использованием данного диска ознакомьтесь с инструкцией, опубликованной в журнале на стр. 109!

Страница 2

Сравнение

Дистрибутивы NAS
EasyNAS 07.2
www.easynas.org
FreeNAS 11.1
www.freenas.org/11.1
NAS4Free 11.1
www.nas4free.org
OpenMediaVault 3.0.94
www.openmediavault.org
Rockstor 3.9.1
Rockstor.com

Все дистрибутивы представляют ISO-образы, который можно записать на отдельный носитель, и загрузить в live-режиме прямо с LXFDVD. У всех присутствует возможность установки на жесткий диск.

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

Дорожные диски В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех старых процессорах, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аппаратный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, предоставленных нашей программой или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, с сайта разработчика узнайте все о вашей системе и драйверах.
Тираж издательства ООО «Марком», 188652, Россия, Ленинградская область, Всеволожский р-н, дер. Юрки, Школьная ул., 7-а. Лицензия ИДТР ВАР № 77-03.

» СОЗДАНИЕ УСТАНОВОЧНЫХ ДИСКОВ ПРИ ПОМОЩИ CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу — это обратиться к программе *cdrecord*. Для всех перечисленных ниже действий потребуются права суперпользователя-root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке — например, 0,3,0. Теперь вы можете с легкостью записать образ на диск: `cdrecord dev=0,3,0 -v /путь к образу/image.iso`

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле `/etc/default/cdrecord`. Добавьте по одной строке для каждого устройства записи (скорее всего, в вашей системе присутствует только одно такое устройство): `Plextor= 0,3,0 12 16M`

Первое слово в этой строке — метка; затем после адреса SCSI-устройства вы должны указать скорость и объем буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Всё будет еще проще, если вы добавите следующее: `CDR_DEVICE=Plextor`

Для записи ISO-образа вам осталось набрать команду `cdrecord -v /path/to/image.iso`

Если вы не принадлежите к любителям командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее от имени root и выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на **Combust!** Пока ваш образ прожигается на диск, можете выпить чашечку кофе.

ДРУГАЯ ОС?

Использовать Linux для записи компакт-диска не обязательно. Программы вроде *cdrecord* просто переносят двоичные данные на чистую матрицу. Все необходимые файлы уже включены в ISO-образ, который распознается любой операционной системой, будь то Linux, Windows, Mac OS X или AmigaOS.

НЕТ УСТРОЙСТВА ДЛЯ ЗАПИСИ ДИСКОВ?

Если у вас нет устройства, с помощью которого можно было бы записать образ на диск, можно найти кого-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 сайт разработчика дистрибутива.

LINUX FORMAT

Подписывайтесь и читайте Linux Format на iPad или iPhone!

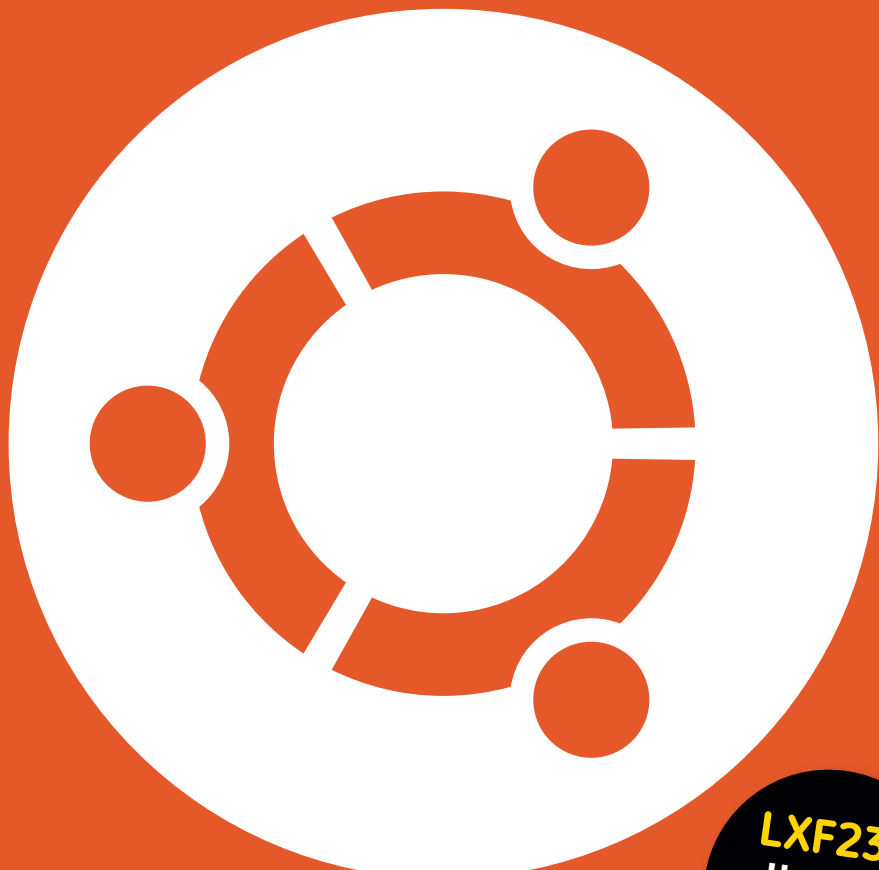
Доступно
в AppStore



А если у вас Android, подпишитесь
на Linux Format через Zinio!

 zinio™
Доступно в Google Play





LXF237
Читайте
в июньском
номере

Готовьтесь к...

UBUNTU 18.04 LTS

Грянул большой релиз Ubuntu — как его получить, как обновить, как настроить и как починить.

Квантовые компьютеры

Для проведения квантовых экспериментов вам не потребуется подземной лаборатории с супер-охлаждением. Мы подскажем, как к ним приступить.

Лучшая CMS

Всемирная Сеть держится на открытом коде; мы рассмотрели наши системы управления контентом и сообщим вам, которая лучше!

Все в контейнеры!

Мир Linux помешался на контейнерах! И эта замечательная новая технология попала под микроскоп LXF.

Компьютерная классика

Оживите былые дни со своим старым Atari ST: мы углубляемся в эмуляцию Atari и TOS с открытым кодом.

Содержание будущих выпусков может меняться — вдруг мы не успеем переформатировать (опять) наши жесткие диски...

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № ФФ77-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж печатной версии 1000 экз., распространение электронной версии 30000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Даниил Кривошеин, Светлана Кривошеина, Валентин Развозжаев, Валерий Смирнов, Елена Толстякова, Ирина Шулакова

Редактор диска

Григорий Огилько

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 «Типографский комплекс „Девиз“» 195027, Санкт-Петербург, Якорная ул., 10, корп. 2, литер А, пом. 44
Заказ ТД-0000

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор (Neil Mohr) neil.mohr@futurenet.com

Робот-технар Джонни Бидвелл (Jonni Bidwell)

jonni.bidwell@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоса

(Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.com

Выпускающий редактор Клифф «Марвин» Хоуп (Cliff 'Marvin' Hope)

clifford.hope@futurenet.com

Директор редакции Грэм Барлоу (Graham Barlow)

Старший художественный редактор Джо Гулливер (Jo Gulliver)

ПОДГОТОВКА МАТЕРИАЛОВ

Тим Армстронг (Tim Armstrong), Матс-Таре Аксельссон (Mats Tage Axelsson), Нейл Ботвик (Neil Botwick), Шон Конвай (Sean D. Conway), Кент Ельчук (Kent Elchuk), Таня Фиш (Tanya Fish), Дэн Фрост (Dan Frost), Ти Джей Хейфер (T J Hafer), Джереми Лэйд (Jeremy Laird), Брайан Морк (Brian Mork), Адам Оксфорд (Adam Oxford), Лес Паундер (Les Pounder), Шашанк Шарма (Shashank Sharma), Валентин Синицын (Valentine Sinityn), Александр Толстой (Alexander Tolstoy), Михалис Цукалос (Mihalis Tsoukalos), Джарред Уолтон (Jarred Walton), Евгений Балдин, Евгений Голышев, Андрей Гондаренков, Владимир Диаконов, Алексей Федорчук, Игорь Штомпель

Иллюстрации Шейн Коллиндж (Shane Collinge)

Иллюстрация с обложки magictorch.com

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Тел. +44 (0) 344 848 2852, 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 журналах, издаваемых Future plc group company, обращайтесь на сайт www.futureplc.com.

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

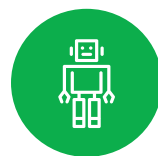
- Праздничные мастер-классы
- Городской лагерь на время каникул
- **Курсы повышения квалификации для педагогов**



Инновационная программа обучения

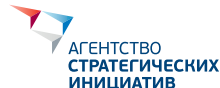


Собственная методическая база



Российские технологии и оборудование

ВООБРАЖАЙ! ПРОГРАММИРУЙ! СОЗДАВАЙ!



www.robboclub.ru

8 800 777 2985

vk.com/robboclub ● info@robboclub.ru

HETZNER
ONLINE

МНОГО ОБЛАКА

ЗА МАЛЫЕ ДЕНЬГИ!

208 руб.



Облачный сервер CX11

Intel® Xeon® Skylake
1 vCPU
2 ГБ оперативной памяти
20 ГБ NVMe SSD
20 ТБ трафика*
Удобная панель Cloud Console
Бесплатная защита от DDoS-атак
Расположен в Германии



208 рублей в месяц

* Все тарифы включают 20 ТБ трафика. Дополнительный трафик оплачивается по 84 руб. за ТБ.

Пришло время доступного облачного хостинга

Наша интеллектуальная панель Cloud Console упрощает использование облачных серверов Hetzner для всех ваших ИТ-потребностей. Она позволяет вам создавать новые серверы менее чем за десять секунд. Для серверов Hetzner Cloud применяется гибкая модель оплаты. У каждого сервера есть месячная цена. Но если сервер использовался менее месяца, то будет применена почасовая оплата.

ru.hetzner.com

Все цены вкл. НДС 18%. Цены могут измениться без уведомления. Все права защищены соответствующими производителями. Intel, логотип Intel, Xeon и Xeon Inside являются товарными знаками корпорации Intel в США и других странах.