

УСКОРИМСЯ | 12 ПРОЕКТОВ

Займемся квантовыми вычислениями!

Бэкап в облаке; создаем гибкие заметки

LINUX FORMAT

Главное в мире Linux

**ДАРИМ
8-ГБ DVD!**
ГОТОВ К ЗАПУСКУ

- UBUNTU 18.04
- ZORIN OS
- SLAX & ЕЩЕ



UBUNTU 18.04 ОСВОИМ!

- Гид по установке и обновлению
- Настройка нового рабочего стола
- Приступим моментально!



Томас Сигдеста́д — об использовании браузеров как операционных систем

Для Pi



ЗАПУСК ЦЕЛОГО ATARI ST

НАЧИНАЕМ С ARDUINO

WINDOWS-ПРИЛОЖЕНИЯ НА ВАШЕМ Pi

ИЮНЬ 2018 LXF237

УЛУЧШИМ САЙТЫ

Лучшие открытые решения для создания сайтов

DOCKER – ЭТО ПРОСТО

Благодаря контейнерам, выпускать и запускать ПО легко!



4 607130 825167 1 8006



iTeleRadio

ИНТЕРАКТИВНОЕ РАДИО&ТЕЛЕВИДЕНИЕ

iteleradio.ru



**LINUX
FORMAT**



» КТО МЫ

На сей раз мы сказали нашим экспертам: вот прибыл Ubuntu 18.04. А вы намерены обновиться? Или нет? Расскажите читателям свои мысли — и предъявите свою работу!



Джонни Бидвелл

Я рисковал здоровьем своей рабочей машины и своими собственными трудами/рассудком, установив на ней дневной образ, который мы поставили на наш диск. Вот на какие жертвы я готов идти, чтобы вы, наши славные читатели, не имели проблем с диском. Я постоянно обновляю образ, и свежий Ubuntu мне очень даже нравится. Но я все равно предпочитаю Arch...

очень даже нравится. Но я все равно предпочитаю Arch...



Ник Пирс

Несмотря на название (Бионический Бобер? Да ну?), я обновлюсь до Ubuntu 18.04. А перед этим сделаю новую резервную копию, поскольку кривые обновления добром не кончаются, и я предпочел бы безопасность плавного отката, а не необходимость сноса или переустановки с нуля.



Лес Паундер

Вышел Ubuntu 18.04, и буду ли я обновляться? Ну да. Мой верный Lenovo X220 гонял 16.04 с момента выпуска, и я созрел принять новое. С наибольшим нетерпением ожидаю новый менеджер файлов *Nautilus*, поскольку провожу в нем немало времени.



Маянк Шарма

Поскольку Нейл не просил меня сделать такой обзор, я собираюсь его пропустить. Новую тему рабочего стола перекинули на осенний выпуск, так что на рабочем столе всё равно будет примерно то же самое.



Шашанк Шарма

По мне, все больше дистрибутивов угождают только своим пользователям. Не вижу смысла пихать в мою установку Arch еще и Ubuntu 18.04. Найти место для виртуальной машины всегда можно, но вряд ли она протянет весь срок типичного цикла выпуска Ubuntu.



Валентин Синицын

Лично я обновился спустя неделю после релиза. Люблю все новое — а новый Wayland в 17.10 как-то не дружил с док-станцией моего ноутбука. Обновился бы и раньше, но не мог: в день выхода 18.04 был вдали от цивилизации, и проблемы с Wayland отошли на второй план.



46 российских ОС



Ровно столько записей было в разделе «Операционные системы» «Единого реестра российских программ для ЭВМ и БД» на момент написания этой колонки. Кроме систем общего назначения, в этот раздел попали операционки для смарт-карт, сетевого оборудования и тонких клиентов, да еще пара гипервизоров. Помимо «давних знакомых», таких как Альт, РОСА или Astra Linux, в реестре удалось найти несколько весьма достойных записей. Итак, рейтинг главного редактора **LXF**:

- » Первое место по праву занимает ОС «Синергия 1.0», разрабатываемая в недрах МинАтома. О ней вообще ничего не известно. Ссылка на страницу разработчика ведет на сайт института, но о проекте там — ни слова. Нагуглить удалось лишь пару сообщений в прессе и совет на каком-то форуме, как подключить к этой ОС репозиторий Debian. Я понимаю, в атомном ведомстве секретность должна быть просто термоядерной...
- » Второе место я присудил «Ульяновск BSD», чей правообладатель (так в реестре) г-н Волков Сергей Вячеславович за весьма умеренную плату предоставляет свою сборку FreeBSD и техническую поддержку.
- » Тройку лидеров замыкает AlterOS от группы компаний «АЛМИ». Программа установки честно «смотрит» в репозиторий openSUSE Leap 42.3, только на сайте авторов об этом скромно умалчивается.

Такое вот импортозамещение...

Но есть и действительно оригинальная разработка — QP ОС, написанная «с нуля» специалистами пензенской компании «Криптософт». Кстати, если вам интересны новые ОС, которые «не Linux», обратите внимание на Sculpt TC (вторая страна DVD этого номера) и проект Genode.

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

Zentyal 5.1 14

Релиз с точкой, пробывший в разработке добрый год, скорее всего, не лишен интереса. Но уместно ли любопытство насчет серверного дистрибутива с коммерческим уклоном?

MX Linux 17.1 15

Почти библейский пример того, что «мал золотник, да дорог». Не исключено, что это самый простой и ровный дистрибутив 2018 г. — по крайней мере, пока.

Intel Optane SSD 800p 16

Тот ли это долгожданный, доступный по цене SSD от Intel? Похоже, что не совсем, из-за интерфейса M.2; но он уж точно доступный, и его ввод и вывод зашкаливают!



AMD Ryzen 3 2200G 17

Четырехъядерный процессор с интегрированной графикой — мечта любого самосборщика игровых компьютеров. Это если суметь заставить его работать на Linux...

Before the Storm 18

Следим за приключениями девочки-подростка, которой не чужды ни романтика, ни вспышки ярости, ни типичные проблемы этого возраста.



UBUNTU 18.04 ОСВОИМ!

Новый LTS-релиз уже с нами! Установим, обновим, настроим и насладимся новыми функциями — всё на с. 28!



СПРАВНЕНИЕ



Системы управления контентом 22

Свив себе уютное гнездо в уголке Интернета, надо его украсить — да так, чтобы оно смотрелось мило на всех устройствах. CMS-то у нас зачем?

ИНТЕРВЬЮ



Беседуем о web-приложениях с **Томасом Сигдестадом**, основателем Eponic 36

ПОЛЬЗОВАТЕЛЯМ RASPBERRY PI



Новости Pi	89
Малиновые поля в расцвете, множатся Jam'ы по всей стране, плюс интригующий проект Музея-в-коробке.	
ExaGear Desktop	90
Запускайте из рабочего стола своего Pi программы Windows. Не бойтесь, в наше время за это уже не осудят.	
Комплект молодого бойца	91
Достойный пакет программ для Pi запускается с ходу, без головной боли, и порадует практически всех.	
Безопасные связи Arduino	92
Raspberry Pi и Arduino легко стать закадычными друзьями и бок о бок работать над совместными проектами.	
Эмуляция Atari ST	94
Путешествуем во времени в прошлое, чтобы преобразить ваш Pi в полноценную машину Atari ST и запустить на ней диски.	



АКАДЕМИЯ КОДИНГА

Генератор сайтов — Hugo	80	Мультисайт на WordPress	84
Создаем безопасные сайты со статическими файлами, используя среду Hugo, которая разработана на Google Go.		Обзаведитесь инструментами создания и клонирования мультисайтов, для более удобного управления ими. Все препятствия отшвырнем!	

ПОСТОЯННЫЕ РУБРИКИ

Новости	6	Рубрика сисадмина	50
«Байкалы» идут в розницу, Tesla у каждого своя, Intel взял рубеж 10 нм, вышли новые SODIMM, Ростех затевает альянс, Micron и Intel сдружились, ZenBook порадовал сенсорной панелью, а Dell верна Linux.		Побеспокоившись о безопасности AMD, под спокойным, уверенным и рассудительным руководством д-ра Валентина Сеницына обучаемся работе с Nftables.	
Вести мобильных ОС	20	Ответы	98
«ВымпелКом» и Huawei развивают 5G, ARM Cortex-A76 ускоряет смартфоны и нетбуки, ИИ пошел в мобильные устройства среднего уровня, а Mobvoi показала «умные» часы.		Заменяем инструментарий MacOS, починаем проверку орфографии, обновляем релиз Ubuntu, разбираемся с настройкой NAS и VPN.	
Сравнение	22	HotPicks	100
Чей сайт краше? Выбираем между системами управления контентом CMS Made Simple, Concrete5, Drupal, Exponent CMS, Joomla.		Отведаете горяченького! Лучшие в мире новинки свободного ПО: <i>Amarok, Battle for Wesnoth, Gnome, Heimer, Krita, Krunner-translator, Qsoundrec, RawTherapee, Shotwell, Stacer, Tower game.</i>	
Интервью LXF	36	Диск Linux Format	106
Томас Сигдестад хочет приносить пользу людям и сосредотачивается на прогрессивных приложениях.		Предыдущие номера	108
		Через месяц	112

ЗАДАРОМ НА DVD

Ubuntu 18.04 'RC'
Zorin OS 12.3
Slax 9.4.0... и не только

с. 106

UBUNTU

Релиз года, без которого буквально не жить!

- Более быстрая и легковесная сборка
- Новый рабочий стол Gnome
- Долгосрочная поддержка
- Обновленное ядро 4.15

Zorin OS 12.3

Идеальное приближение для пользователей Windows и MacOS

Slax 9.4.0

Крошечный и мощный: эту ОС можно запускать с USB-брелка

УЧЕБНИКИ

ТЕРМИНАЛ: Moreutils	56
Малоизвестная коллекция утилит Unix увеличит продуктивность вашей работы в терминале.	
CHERRYTREE: Организуемся	58
Внедрите свои умные мысли в потрясающие проекты, предварительно разложив их по полочкам.	
LEDE: WordPress в кармане	60
Среда LEDE, дешевый роутер, чуть-чуть изобретательности — и готов ваш собственный сервер разработчика.	
БЭКАП: Личное облако	64
Запустим себе облако с процедурами резервного копирования и восстановления, при максимуме безопасности и удобства.	
DOCKER: Виртуальные приложения . . .	68
Забываем про виртуальные машины и изучаем виртуальные приложения. Глядишь, экономия.	
ТУЛЧЕЙНЫ: Сборка Linux	72
Стоит ли мелочиться с отдельными инструментами? Соберем сразу целую операционную систему!	
DOCKER: Поваренная книга	76
Предлагаем пару-тройку полезных рецептов изготовления образов Docker.	
Квантовые вычисления	42
Даже простые смертные сумеют работать с буквально самой крутой в мире технологией.	
Devuan и его предки	46
Правы ли systemd-ненавистники? Поинтересуемся развитием альтернативной системы.	

УГЛУБЛЕННО...

Новости

В ЭТОМ НОМЕРЕ: «Байкалы» — всем » Tesla, но не та » Intel уже с 10 нм » Новые SODIMM » Альянс Ростех » Micron и Intel » Панель-инновация » Dell на Linux

ИМПОРТОЗАМЕЩЕНИЕ

«Байкал» в розницу

Российские процессоры BE-T1000 поступили в розничную продажу.

А О «Байкал Электроникс», российская фаблес-компания и производитель микропроцессоров, объявила о начале розничных продаж своей продукции в количествах, необходимых для прототипирования и производства тестовых образцов продукции.

С 1 июня в интернет-магазине электронных компонентов «ЧИП и ДИП» и розничной сети магазинов можно приобрести отечественные системы-на-кристалле BE-T1000 (кодовое наименование Байкал-T1). Розничная цена одного процессора составляет 3990 руб.

Снк BE-T1000 соответствует требованиям к промышленной продукции российского производства и классифицирована в качестве интегральной схемы 2-го уровня.

«Мы рады расширить сотрудничество с одной из крупнейших международных розничных сетей.

Стартовым проектом для нас стала розничная продажа отладочных плат ВФК3.1 — мы дали всем желающим возможность наглядно в деле проверить и использовать технические возможности и высокую производительность отечественных процессоров, — говорит Константин Щербаков, директор департамента маркетинга и продаж АО «Байкал Электроникс», — Мы получили хороший результат, который понравился и нам, и нашим партнерам. С момента выпуска промышленной партии процессоров мы провели большую работу и сейчас можно констатировать, что с точки зрения экосистемы у нас всё готово для поддержки проектирования: от лаборатории на базе ВМК МГУ, где можно «попробовать процессор на вкус», до простой покупки процессора и отладочной платы для создания прототипов устройств».



Российские процессоры «Байкал» впервые поступили в розничную продажу не в составе оценочных плат (одноплатных компьютеров), а в качестве самостоятельных товарных единиц.

НРС для БИЗНЕСА

В партнерстве с мировым лидером

REG.RU предложил новую услугу для ИИ-проектов на базе NVIDIA Tesla V100.

О течественный хостинг-провайдер REG.RU предложил услугу «Облачные вычисления на GPU» (графических процессорах), дающую пользователям доступ к аппаратной платформе для работы с алгоритмами глубокого обучения и высокопроизводительными вычислениями. Кроме того, платформа NVIDIA отлично подходит для 3D- и видеорендеринга. В основе услуги — наиболее мощный специализированный ускоритель для работы с искусственным интеллектом и высокопроизводительных вычислений, доступный на рынке в данный момент: NVIDIA Tesla V100. В рамках услуги можно подключить к одному виртуальному контейнеру до 8 вычислительных ускорителей либо арендовать физический сервер целиком. «REG.RU выбрал в качестве

стратегического партнера мирового лидера по разработке GPU — NVIDIA. Вместе мы хотим развеять миф о том, что новые технологии — это привилегия только гигантов бизнеса и крупных корпораций. Наша цель — дать студентам, преподавателям, ученым, институтам, стартапам, микро-, малому и среднему бизнесу возможности реализовать свои идеи на современных и при этом финансово доступных аппаратных платформах», комментирует гендиректор REG.RU Алексей Королюк. Оплата услуги — посуточная или помесечная. Стоимость одного ускорителя Tesla V100 — от 2,5 тыс. руб. в сутки или 80 тыс. руб. в месяц — зависит от потребностей клиента. Бесплатно предлагается тестовый период до нескольких дней (до 2-х ускорителей к одному контейнеру).

НА РУБЕЖЕ 10 НМ

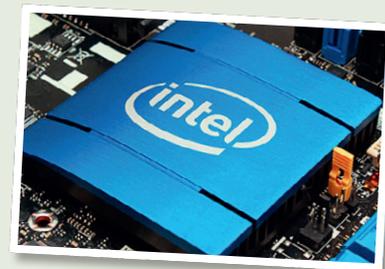
Лучше поздно, чем никогда

Cannon Lake — первый чип Intel, изготовленный по техпроцессу 10 нм.

Выпуск первого «камня» Intel по техпроцессу 10 нм планировался на конец 2016 г., но из-за технических проблем, возникших при уменьшении размеров транзисторов в таких масштабах, был отложен до 2017 г. В своем финансовом отчете за I квартал компания признала, что намеченные объемы производства переносятся со второй половины 2018 г. до неопределенного момента в 2019 г., что вызвано «проблемами с доходностью». Однако в середине мая компьютерное онлайн-издание AnandTech отметило наличие первого 10-нм чипа Intel, Core i3-8121U, в доступном в ритейлеров Китая ноутбуке Lenovo Ideapad 330, а Intel предоставила подробную информацию об этом процессоре в своем каталоге ARK. Итак, производитель официально подтвердил, что Core i3-8121U — процессор архитектуры Cannon Lake и выполнен по технологическому процессу 10 нм. Элементы «i3» и «8» в его названии сообщают, что это представитель 8-го поколения процессоров начального уровня (подобно Kaby Lake-R, Kaby Lake-G и Coffee

Lake), предназначенный для ноутбуков среднего уровня. Согласно спецификации, i3-8121U — 2-ядерный процессор с четырьмя потоками, базовой тактовой частотой 2,2 ГГц (3,2 ГГц в режиме Turbo Boost), кэшем 4 МБ, TDP 15 Вт, поддерживающий до 32 ГБ памяти. При этом Cannon Lake способен работать с двумя новыми видами памяти, LPDDR4 и LPDDR4X (разновидности DDR4 с малым энергопотреблением). В данных, предоставленных Intel, нет описания интегрированной с этим чипом графики. Остается думать, что таковой у нового чипа просто нет. Предположение подкрепляется тем фактом, что Lenovo Ideapad 330 для Китая стал первым устройством с CPU начального уровня, оснащенным дискретным графическим чипом AMD.

Core i3-8121U — чип для Intel не самый яркий, но, безусловно, знаковый: наконец-то выпущен продукт по технологии 10 нм. Причем конкуренты ушли далеко вперед: AMD планирует запуск II и III поколения микропроцессоров Zen по 7-нм технологии, а TSMC уже начала производство своего первого 7-нм «камня».



Изготовленные по техпроцессу 10 нм чипы Intel Cannon Lake замечены в ноутбуке среднего уровня Lenovo IdeaPad 330.

КОМПЬЮТЕРНАЯ ПАМЯТЬ

Для геймеров и не только

Samsung начинает массовое производство новых SODIMM-модулей памяти на основе 10-нм технологии.

Samsung Electronics начала первое в индустрии массовое производство 32-ГБ модулей оперативной памяти DDR4 для игровых ноутбуков, в форм-факторе двусторонних. Новые модули SODIMM, на технологии 10-нм класса, позволят геймерам наслаждаться на ноутбуках играми с качеством, близким к настольному ПК.

На новых модулях возможна разработка более производительных игровых ноутбуков с батареей повышенной емкости и большим объемом памяти при сохранении существующих размеров. Как отметил старший вице-президент по маркетингу систем памяти, г-н Севон Чан [Sewon Chun], 32-ГБ модули памяти DDR4 DRAM от Samsung обеспечат более полное погружение в игровой процесс на ноутбуках: «Мы продолжим наполнять наши новые портфолио DRAM-модулей памяти более быстрыми и емкими образцами для всех основных сегментов рынка, включая премиальные ноутбуки и настольные ПК».

По сравнению с 16-ГБ модулями памяти SODIMM на основе 8-гигабитных микросхем DDR4 с 20-нм

технологией 2014 г., новый 32-ГБ модуль обладает удвоенной емкостью, на 11% быстрее и потребляет на 39% меньше энергии в активном режиме. С 16-ю новейшими 16-ГБ чипами (по 8 чипов на каждой стороне модуля) 32-ГБ модуль SODIMM позволяет увеличить частоту передачи данных до 2666 Мбит/с. Ноутбук с 64 ГБ ОЗУ (два 32-ГБ модуля DDR4) потребляет менее 4,6 Вт в активном режиме и менее 1,4 Вт в режиме ожидания, что на 25% энергоэффективнее существующих 16-ГБ решений. Samsung приступил к активному развитию новой линейки модулей памяти DRAM на технологии 10-нм класса (16 ГБ LPDDR4, 16 ГБ GDDR5 и 16 ГБ DDR4), открыв новую эру 16-гигабитных модулей памяти DRAM в мобильных устройствах, графических подсистемах, ПК и серверах, и распространит их на суперкомпьютеры и автомобильную электронику.

НОВЫЕ МОДУЛИ ПОЗВОЛЯТ ГЕЙМЕРАМ НАСЛАЖДАТЬСЯ НА НОУТБУКАХ ИГРАМИ КАК НА ПК.

РОССИЙСКОЕ ПО

На горизонте — экосистема «Эльбруса»

Ростех создаст альянс разработчиков программного обеспечения для компьютеров «Эльбрус».

Структуры Ростеха — Национальный центр информатизации и Институт управляющих машин им. И. С. Брука — планируют создать альянс разработчиков ПО для отечественных процессоров «Эльбрус», до конца 2018 г. присоединив не менее 20 компаний-разработчиков, а скачать приложения для «Эльбруса» можно будет с удобного маркетплейса, аналогичного Google Play или AppStore. Об этом было объявлено в рамках конференции ЦИПР-2018.

Альянс создается в целях широкомасштабного внедрения решений на базе российской микроэлектроники. Предполагается, что с выходом значительного объема совместимого ПО появятся новые качественные аппаратные платформы — материнские платы, серверы, рабочие станции, промышленные ПК, что, в свою очередь, снизит конечную стоимость решений. Результатом выстраивания экосистемы на процессорах «Эльбрус» станет рост доли рынка российских процессоров, внедрение безопасных, доверенных аппаратных систем.

ИНЭУМ и НЦИ разработали дорожную карту развития проекта — не менее трех пилотных внедрений программно-аппаратных комплексов на базе «Эльбрус» с готовым прикладным ПО уже до конца 2018 г. Помимо компаний-разработчиков, к разработке ПО планируется привлечь студентов и аспирантов лучших вузов России.

Так студенты получают возможность не только применить и расширить свои знания, но и коммерциализовать

результаты работы. «Развитию и внедрению собственной электронной компонентной базы в мире уделяется огромное внимание. Россия не может остаться в стороне. Барьер на пути массового применения отечественных процессоров — отсутствие прикладного ПО», сказал гендиректор НЦИ Константин Солодухин.

СТУДЕНТЫ ПОЛУЧАТ ВОЗМОЖНОСТЬ НЕ ТОЛЬКО ПРИМЕНИТЬ И РАСШИРИТЬ СВОИ ЗНАНИЯ.

ФЛЭШ-ПАМЯТЬ

Терабитный чип вместо HDD

Micron и Intel укрепляют лидерство в области технологий памяти 3D NAND.

Компания Micron Technology Inc. и корпорация Intel объявили о запуске в производство и начале поставок первых в отрасли микросхем памяти 3D NAND с четырьмя битами на ячейку. Построенная на базе уже зарекомендовавшей себя 64-слойной структуры, новая технология NAND памяти с записью четырех бит на ячейку обеспечивает плотность записи в 1 Тбит на кристалл, что является самой высокой на сегодняшний день плотностью записи для флэш-памяти.

Кроме того, компании сообщили о новых успехах в области создания 96-слойной структуры 3D NAND памяти третьего поколения, в которой число слоев увеличено на 50% по сравнению с предыдущим поколением. Эти достижения в структуре ячеек позволяют компаниям сохранить лидирующие позиции в производстве микросхем с самыми высокими показателями плотности записи.

Оба технологических достижения в области NAND памяти — 64-слойная QLC и 96-слойная TLC память —

используют технологию CuA (CMOS under array), что позволяет сократить размер кристалла и обеспечить улучшенную производительность при сравнении с другими конкурирующими технологиями. За счет использования четырех плоскостей по сравнению с двумя у конкурентов, новая флэш-память NAND производства Intel и Micron позволяет осуществлять запись и чтение одновременно на большем числе ячеек, что приводит к увеличению скорости записи и повышению пропускной способности на уровне всей системы.

Новая 64-слойная технология NAND памяти с записью четырех бит на ячейку позволяет добиться более высокой плотности хранения данных в меньшем объеме, обеспечивая значительную экономию для рабочих нагрузок в облачных окружениях с большим числом операций чтения. Она также отлично подходит для потребительских и клиентских вычислений, обеспечивая оптимальные с точки зрения затрат решения для хранения данных.



Micron 5210 ION, первый SSD на базе новой флэш-памяти.

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

НОУТБУКИ

А два экрана — лучше!

ASUS ZenBook Pro 15 — ноутбук с инновационной сенсорной панелью.

Главная особенность представленного на Computex 2018 ноутбука ASUS ZenBook Pro 15 — встроенный в его тачпад дополнительный 5,5-дюймовый IPS дисплей с разрешением Full HD (1920×1080). Многофункциональный тачпад, названный производителем ScreenPad, способен превращаться в цифровую клавиатуру, калькулятор или панель запуска приложений. ScreenPad также может выступать в роли дублера основного экрана или использоваться для просмотра видео на YouTube. ASUS интегрировал в новинку пакет *Microsoft Office*: на ScreenPad работают ярлыки *Word*, *PowerPoint* и *Excel*; в дальнейшем потенциал ScreenPad сделают доступным и другим сторонним приложениям. Начинка ZenBook Pro 15 состоит из CPU VIII поколения Intel Core i9, i7 и i5, до 16 ГБ ОЗУ, HDD до 1 ТБ или PCIe SSD до 512 ГБ, и графика Nvidia GeForce GTX 1050 Ti. Хороший набор портов, включающий USB Type-A и Type-C с Thunderbolt 3, время автономной работы от батареи — 10 часов. Основной дисплей ZenBook Pro 15 обеспечивает 100% охват цветового пространства Adobe RGB, 132% — sRGB,



Ноутбук с двумя экранами ZenBook Pro 15 — главная новинка компании ASUS на Computex 2018.

при значении Delta E менее 2. Продажи начнутся в середине июля, стартовая цена составит \$2299 (около £1700). Цветовых решений два: синий и «розовое золото» с блестящими гранями.

Кроме того, будет предложена модель ZenBook Pro 15 с меньшим дисплеем (14-in) и более низкими спецификациями — но и, соответственно, с более доступной ценой.

CDELLANO DELL

Precision значит Точность!

CPU Coffee Lake + Ubuntu Linux = мобильная рабочая станция Precision.

Dell предлагает обновленную линейку ноутбуков Precision (3530, 5530, 5530 2-в-1, 7530 и 7730) с чипами Intel Coffee Lake VIII поколения и дистрибутивом Ubuntu на борту. Рабочая станция начального уровня Dell Precision 3530 поставляется с Ubuntu Linux 16.04 и процессором Intel Core i5-8400H, до 32 ГБ DDR4 ОЗУ, графика — интегрированная или дискретная карта с GPU Nvidia Quadro P600. Мобильные рабочие станции Dell Precision 7530 и 7730 позиционируются как модели 'Developer Edition', и комплектуются процессорами Coffee Lake, графикой NVIDIA Quadro P3200 или AMD Radeon Pro WX 4150 и дисплеями с диагональю 15" и 17". Все эти бизнес-ориентированные ноутбуки могут быть укомплектованы процессорами Intel Xeon; кроме того, объем ОЗУ на старшей модели 7730 — до 64 ГБ, а накопителей до 8 ТБ. Поддержка интерфейса Thunderbolt 3 позволяет последовательно подключить несколько мониторов 4K. Dell уже не раз выпускала свои ноутбуки с предустановленной Ubuntu. Начало доброй традиции положил

проект Sputnik в конце 2012 г., когда для разработчиков компании вышла версия "developer edition" ноутбука XPS 13. Обновленная в 2018 г. модель XPS 13 опционально доступна в корпусе белого цвета, а также новую версию предустановленной Ubuntu.



Компания Dell обновила линейку своих мобильных рабочих станций Precision.

Короткой строкой

➤ Начиная с релиза Kubuntu 18.10, для загрузки будут предлагать установочные образы только для архитектуры x86_64. Источник: www.omgubuntu.co.uk

➤ Проект Fusion, совместно основанный разработчиками Tor и Mozilla, нацелен на интеграцию поддержки Tor в Firefox. Источник: news.ycombinator.com

➤ Компании Huawei предложила новую файловую систему EROFS (Extendable Read-Only File System), для разделов, доступных только в режиме read-only. Источник: lkm1.org

➤ Компания Samsung объявила об открытии в России Центра искусственного интеллекта, где будут изучаться компьютерное зрение и базовые алгоритмы. Источник: news.samsung.com

➤ Проект TrueOS (ранее PC-BSD) объявил о превращении из надстройки над FreeBSD с графическим рабочим столом в обособленную ОС ветку FreeBSD. Источник: www.trueos.org

➤ Microsoft приобрела хранилище программ с открытым кодом GitHub за \$7,5 млрд. Возглавит новое подразделение вице-президент Microsoft Нат Фридман [Nat Friedman]. Источник: www.cnews.ru

➤ В ядре 4.17 удалено 8 устаревших процессорных архитектур, добавление в XFS опции lazytime, полная реализация протокола TLS на стороне ядра, защита от уязвимостей Spectre 3a/4. Источник: www.opennet.ru

➤ Согласно рейтингу Net Applications, доля Firefox за год снизилась с 12,63% в июне 2017 г. до 9,92% в мае 2018 г., а Chrome за год повысил свое присутствие с 60,08% до 62,85%. Источник: www.opennet.ru

➤ Госдума приняла закон о размере штрафов, которые будут платить провайдеры хостинга, если они не раскрывают Роскомнадзору данные о владельцах прокси и VPN. Источник: www.cnews.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.

» ШУТКИ BIONIC'A

Вот уже 5 лет в качестве основной рабочей системы применяю Ubuntu и ее дериваты. И каждый раз перед очередным релизом ее я смотрю: а чего в ней в очередной раз поломали? Ибо до сих пор это происходило с неизбежностью уравнений небесной механики: то раскладку клавиатуры испортят, то кириллицу в консоли. Что быстро исправлялось — но каждый раз вносило сложности в сборку Cuntu, которая тоже принадлежит к отпрыскам Ubuntu, хотя и незаконным.

Правда, перед выходом релиза 18.04, известного как Bionic Beaver, ничто такого поломания не предвещало — Cuntu на базе предыдущего релиза (17.10) собралась на удивление гладко. Однако тут подстерегала засада, связанная с нарушением некоторых зависимостей от *gksu*. Который оказался изъятым не только из штатного комплекта дистрибутива, но и из репозитория вообще. Что произошло буквально за две недели до релиза и обосновывалось его устарелостью. Устарелость же вызвана была неработоспособностью этой команды в *Wayland*'е. Сеанс которого, по парадоксальной логике разработчиков, был исключен из 18.04 по умолчанию — но должен был быть включен в релиз 18.10.

Конечно, это священное право майнтейнеров — считать устаревшим тот или иной компонент системы. Однако помимо из репозитория пакет, от которого зависит множество сторонних пакетов, да еще делать это накануне релиза... на это можно отреагировать только словами пионера-имярек из письма его в редакцию журнала «Костер». Причем «сломанные», опять же совсем недавно, зависимости оказались и у некоторых других пакетов, входящих во вполне официальные репозитории.

Впрочем, это ничуть не помешало своевременному выходу Cuntu 18.04 и даже имело некоторые положительные следствия, о чем будет говорить в следующей колонке.
alv@posix.ru

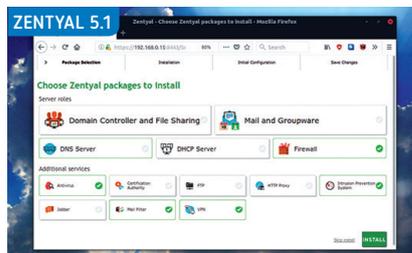
Сегодня мы рассматриваем:

- Zentyal 5.1** 14
Главным новшеством этого релиза является упрощенная работа по автоматическому обновлению заданий *cron* и новые слайды, которые помогают выбирать пакеты для установки.
- MX Linux 17.1** 15
Дистрибутив, идеальный и для новичков, и для бывалых линуксоидов, демонстрирует чудеса скорости (благодаря рабочему столу *Xfce*), практичное ПО и обстоятельную документацию.
- Intel Optane SSD 800p** 16
Первый основной SSD для ПК от Intel на базе технологии 3D XPoint не подтверждает объявленных претензий... да, похоже, вовсе и не собирался этого делать.



Ryzen 3 2200G удивляет своей мощностью при низкой цене.

- AMD Ryzen 3 2200G** 17
С этим процессором вы получаете от 30 до 60 кадров в секунду во вполне современных играх с разрешением 1080p и адекватными настройками, причем по умеренной стоимости.
- Before the Storm** 18
Игра довольно достоверно отражает подростковые комплексы, фантазии, тревоги и тайные желания. Еще бы и физика была столь же достоверная...



Программ по умолчанию показалось мало? Добавьте себе еще.



Герои игры поразительно узнаваемы — таких ребят мы встречаем каждый день.

» СРАВНЕНИЕ: CMS C. 22

Сайт в наше время создать проще простого. Но как добиться, чтобы он выделялся на фоне общей массы? Предлагаем вам удобные инструменты для обустройства сайтов...

Zentyal 5.1

Знаковые версии, которые разрабатываются год — веская причина разжечь интерес **Шашанка Шармы**. Неужели его любопытство неуместно?

ВКРАТЦЕ

В отличие от большинства других серверных дистрибутивов, Zentyal имеет графический рабочий стол, пригодный для точной настройки системы. Его фишкой, однако, является простой в навигации web-интерфейс для установки ПО и настройки различных сервисов. В дополнение к изданию сообщества проект также создает коммерческий вариант, на базе сервера Ubuntu. Очень похожи ClearOS, NethServer и Univention, они тоже имеют web-интерфейс.

С текущим состоянием большинства современных настольных Linux-дистрибутивов, трудно представить времена, когда при настройке системы требовалось вручную задавать горизонтальную и вертикальную частоту обновлений для монитора. Если настройки были указаны неверно, возник серьезный риск сжечь монитор.

С тех пор для настольных дистрибутивов многое изменилось — пользователи могут завершить установку за минуты; но использование Linux для запуска на серверах еще остается более сложным процессом. К счастью, проекты подобные Zentyal, призваны обеспечить механизм слияния многофункционального сервера при наименьшем прибавлении седых волос.

Установка довольно проста и требует лишь немного больше, чем выбор места для дистрибутива и предоставления ключевых данных, таких как раскладка клавиатуры и часовой пояс. Можно настроить дистрибутив с рабочего стола или через web-интерфейс.

Web-интерфейс проведет вас по модулям установки, которые вы захотите запустить на своем Zentyal сервере; это, например, FTP, почтовый сервер, почтовый фильтр, антивирус, система обнаружения вторжений и другие. После этого вы попадаете на Панель управления, откуда можно контролировать все элементы сервера.

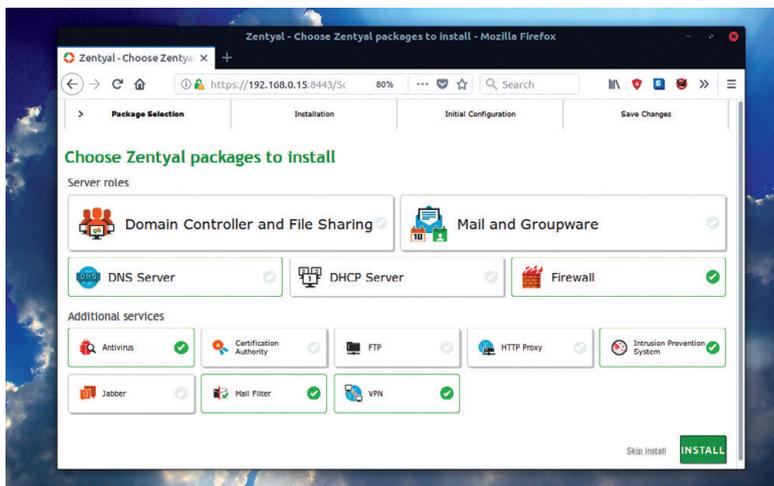
Боковая панель слева от web-интерфейса будет автоматически расти, размещая установленное ПО. Обновления легко устанавливаются одним щелчком мыши. Но учтите, что обновления для сообщества менее тщательно протестированы, чем поставляемые в коммерческом варианте. Проект рекомендует использовать коммерческий вариант, который включает техническую поддержку и предлагает для производственных серверов обновления сроком на 4,5 года.

И я про тоже

Спустя год разработки, вышел последний выпуск, основанный на Ubuntu 16.04.4 LTS и представляющий множество исправлений ошибок, улучшение удобства использования и производительности, а также новые функции. Главным среди этого является упрощенная работа по автоматическому обновлению заданной *scop* и новые слайды, которые помогут вам выбирать пакеты для установки. Журнал изменений проекта содержит полный список изменений в различных компонентах дистрибутива.

Существующие пользователи Zentyal смогут перейти на последнюю версию одним щелчком мыши, а проект работает на пакетах для Zentyal 5.0, 4.2, 4.1 и 4.0, снабженных кнопкой обновления самой Панели управления.

Хотя проект дает доступ к рабочему столу и терминалу, web-интерфейс спроектирован как отличный помощник в работе с сервером. Можно даже выудить журналы для отдельных компонентов, например, *Samba* и *Mail*, и даже отфильтровать их



В дополнение к бесчисленным серверам можно использовать Zentyal для развертывания экземпляра Jabber, например.

до определенной даты — со всеми удобствами браузера. Там, где это возможно, интерфейс позволяет большему количеству элементов еще больше сузить журналы, ограничивая их конкретным интерфейсом или протоколом при просмотре сетевых журналов, например.

Однако эта функция также выявила одну из ошибок, с которыми мы столкнулись при тестировании Zentyal. Независимо от того, какие параметры мы использовали, мы не могли заставить дистрибутив вытягивать любые журналы для отображения. Но кроме этой небольшой ошибки, дистрибутив работает как заявлено.

При всей простоте его развертывания, самое поразительное в Zentyal — это его самопредназначение для малого и среднего бизнеса [SMB], которое сужает функциональность дистрибутива. Например, его нельзя использовать для развертывания web-сервера или даже файлового менеджера без долгой ручной настройки. По нашему мнению, это сводит на нет быстроту и простоту развертывания его самого. **LXF**

ВЕРДИКТ

РАЗРАБОТЧИК: Zentyal S. L.

САЙТ: www.zentyal.org

ЛИЦЕНЗИЯ: Бесплатно (разработка)

ФУНКЦИОНАЛЬНОСТЬ	8/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
УДОБСТВО В РАБОТЕ	9/10
ДОКУМЕНТАЦИЯ	9/10

У Zentyal прочная поддержка по части инструментов SMB, но случайные пользователи могут предпочесть нечто вроде NethServer.

➤ **Рейтинг 8/10**

MX Linux 17.1

Представления **Шашанка Шармы** о современном дистрибутиве подобны грозному Голиафу; дистрибутив поэтому мчится напрямик в лагерь Давида.

ВКРАТЦЕ

Дистрибутив основан на Debian Stable, и можно проследить его происхождение от сообществ Mepis и antiX. Наряду с легкой и отзывчивой средой Xfce, фишка MX Linux — удобство пользователя и множество базовых и расширенных пользовательских инструментов. Если вам нравятся дистрибутивы, поставляемые с инструментами пользователя, то вам также понравится Solus.

Согласно некоторым исследованиям, человеческий мозг гораздо быстрее формирует отрицательные выводы, чем позитивное мнение. Этот феномен можно проверить на себе, попробовав новое пиво или сходяв на очередную фильм Николаса Кейджа. Если вам не терпится, то можно узнать мысли критиков о MX Linux, пробежав взглядом врезку «Вердикт».

Последняя версия MX основана на Debian Stable, хотя проект возник при слиянии сообществ Mepis и antiX. Но не позволяйте его родословной затуманить ваше суждение. Дистрибутив предоставляет установочные ISO-образы для 32- и 64-разрядных машин. Наряду с рабочим столом Xfce, дистрибутив предлагает широкий выбор приложений по умолчанию, включая основное ПО для офиса и мультимедиа и даже несколько игр.

Установщик запускается щелчком по иконке на рабочем столе. Дистрибутив предусматривает собственный установщик, который задает надлежащие вопросы, например, о местоположении дистрибутива, и если вы распорядитесь вырезать место для MX Linux, то установщик запустит *fdisk*. Также надо выбрать пункт назначения для загрузчика *Grub*, но на том и всё.

Настройка в изобилии

Рабочий стол Xfce имеет нижнюю панель и боковую панель, где расположены важные элементы, такие как регулятор громкости и индикатор батареи. В панели запуска размещена панель поиска, которая поможет вам перейти в конкретное приложение без навигации по различным категориям, типа Office, Multimedia и т. д.

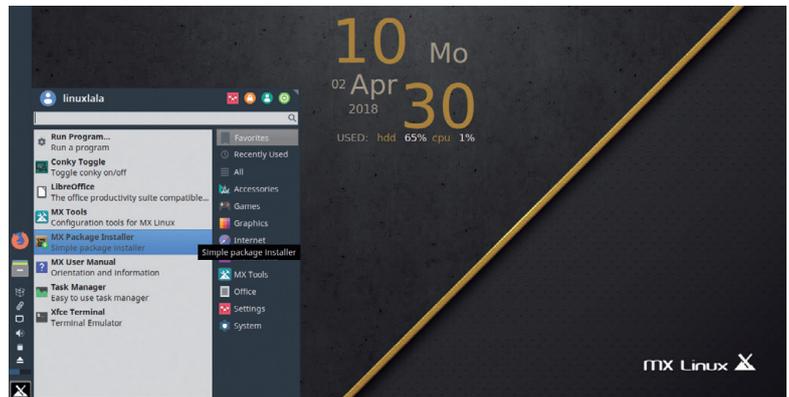
Один из приятных итогов употребления Xfce и, возможно, смешанного наследия, состоит в том, что MX Linux быстр — гораздо быстрее большинства настольных дистрибутивов. Мы глазам своим не поверили, когда нам удалось запустить инструмент *Snapshot* без ошибок даже в сильно ограниченной среде *VirtualBox* при ОЗУ менее 2 Гб.

Еще одно преимущество выбора Debian Stable в качестве основы в том, что MX Linux невероятно надежен. Даже если вы возитесь с малораспространенными инструментами, такими как *SM Tube*, служащем для поиска видео на YouTube. Дистрибутив также поддерживает по умолчанию воспроизведение различных аудио- и видеоформатов.

Имеется также несколько важных пользовательских инструментов, которые официальное руководство относит к продвинутым и базовым. Последний состоит из таких инструментов, как установщик пакетов, установщик графических драйверов NVIDIA и т. д.; в свою очередь, сложные инструменты, типа *Snapshot*, с помощью которых можно создать загрузочный ISO-образ текущей установки, идентифицируются как расширенные.

С помощью инструмента *MX Tweak* можно изменить внешний вид, включая тему и положение панели.

На *MX Package Installer* установленные приложения выделены серым цветом; можно использовать выпадающий список сверху



Среди его многочисленных пользовательских инструментов – встроенный инструмент создания USB-устройств, заимствованный у antiX, и его можно использовать для создания USB-устройства с постоянным хранилищем.

справа, чтобы вывести Установленные, Обновленные и Неустановленные приложения. Приложения отсортированы по соответствующим категориям, таким как Games, Graphics, Office и Torrent. Все эти специальные инструменты и многое другое перечислены в меню MX Tools.

В отличие от большинства других настольных дистрибутивов, MX Linux имеет полное оффлайн-руководство, охватывающее более 100 страниц, а в проекте также размещаются доска объявлений и пользовательские видеоуроки, которые помогут вам акклиматизироваться в дистрибутиве. При наличии видеоуроков, руководство пользователя предусмотрительно помещает на них ссылки.

Обстоятельная документация и ПО, установленное по умолчанию, делают MX Linux идеальным и для новичков, и для бывалых пользователей Linux. Если вы ищете быстрый и надежный дистрибутив, но не хотите переходить на Arch или другой вариант с плавающим релизом, то вряд ли найдете более разносторонний вариант, чем MX Linux. **LXF**

ВЕРДИКТ

РАЗРАБОТЧИК: MX Linux Team
САЙТ: mxlinux.org
ЛИЦЕНЗИЯ: Разные

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

Быстрый, полнофункциональный дистрибутив, в комплекте с надежной поддержкой и в автономном, и в онлайн-режиме; сущий мастер-класс по удобству использования.

» Рейтинг 10/10

Intel Optane SSD 800P

Революционная технология памяти Intel 3D XPoint попадает в мейнстрим.

Джереми Лэйрд пытается сделать хорошую мину.

СПЕЦИФИКАЦИИ

- **Емкость** 58 ГБ, 118 ГБ
- **Формат** M.2
- **Порт** PCIe v3 x2
- **Протокол** NVMe
- **Логика** Intel
- **Тип** Intel 3D XPoint
- **Потребляемая мощность** 3,75 Вт в работе, 8 мВт в простое
- **Долговечность** 365 ТБ на запись
- **Гарантия** 5 лет
- **Габариты** 22×80 мм

У нас в руках первый основной Intel SSD для ПК на базе технологии 3D XPoint (см. **LXF232**) — Intel Optane SSD 800P. Но что это? Последовательное чтение 1450 МБ/с? Запись всего 640 МБ/с? AIOР всего 250К для чтения и 140К для записи? Это значительно слабее предрелизных маркетинговых посулов. Что происходит? Сложно сказать... Мы обнаружили, что 800P по обычным стандартам SSD ничем не примечателен, и это серьезное разочарование, после шумихи, сопровождавшей 3D XPoint. Последовательная пропускная способность вполне соответствует заявлениям Intel и, таким образом, далека от самых быстрых SSD на базе NAND.

Разумеется, Intel выставила для 800P двухполосный интерфейс PCI Express, который ограничивает пиковую производительность передачи по сравнению с NAND-накопителями высокого класса с четырехполосными интерфейсами, но явно не это сдерживает в тестах скорость последовательной записи 800P максимумом в 650 МБ/с — едва ли быстрее, чем SSD с SATA.

Обещание лучшего

Однако метрики по случайному доступу 4K указывают на большой потенциал 800P. 800P показывает 229 МБ/с для чтения и 176 МБ/с для записи. Эти цифры значительно выше, чем может выдать текущий золотой стандарт среди NAND SSD — Samsung 960 Pro. Вообще говоря, основное внимание в этом накопителе уделяется сокращению задержки и росту производительности при малой глубине очереди, а не пропускной способности при пиковой нагрузке.

Более конкретно, Intel говорит, что 800P в четыре-пять раз быстрее реагируют на короткие глубины очереди из одного и двух по сравнению с SSD на основе NAND. Intel считает, что именно очереди небольшой глубины в основном и определяют произ-



800P — первый основной Optane SSD, достаточно большой (просто) для использования в качестве загрузочного диска.

водительность типичных клиентских нагрузок ПК. Intel также заявляет, что производительность поддерживается независимо от заполненности диска, и оценивает 800P в 365 ТБ записей, а это невероятно высоко.

При этом, новый 800P имеет и плюсы, и минусы. Ограниченная емкость означает, что даже больший из двух доступных дисков не слишком подходит в качестве загрузочного. Также обидно видеть, что диск на основе якобы революционной технологии памяти показывает столь посредственную производительность. Как следствие, на данный момент необходима оценка преимуществ этой новой технологии. Это, скорее всего, самый отзывчивый основной SSD, покупаемый за деньги. А также, возможно, самый надежный. Но отнюдь не самый быстродействующий из всех.

Степень разочарования, вероятно, зависит от того, насколько вы купились на рекламу 3D XPoint. В реальности диск отнюдь не подтверждает объявленных претензий, и, похоже, изначально и не собирался это делать. Это явно не мгновенная революция, на какую мы надеялись. **LXF**

»» ОПТАНЕ ПРОТИВ КОНКУРЕНТОВ

Тест	Intel Optane 800P 118 ГБ	Samsung 960 Evo 250 ГБ	Intel 760P 512 ГБ
30 ГБ внутреннее копирование (с)	76	89	66
Последовательное чтение (МБ/с)	1453	3019	3048
Последовательная запись (МБ/с)	650	1494	1619
4К чтение (МБ/с)	229	64	75
4К запись (МБ/с)	176	191	218

ВЕРДИКТ

ПРОИЗВОДИТЕЛЬ: Intel
САЙТ: www.intel.com/optane
ЦЕНА: £ 142 (118 ГБ), £ 85 (58 ГБ)

ФУНКЦИОНАЛЬНОСТЬ	8/10
ПРОИЗВОДИТЕЛЬНОСТЬ	7/10
УДОБСТВО В РАБОТЕ	9/10
ОПРАВДААННОСТЬ ЦЕНЫ	7/10

Сверхнизкая латентность и повышенная долговечность пасуют перед высокой ценой и средней пропускной способностью.

»» **Рейтинг 7/10**

AMD Ryzen 3 2200G

Вдруг это ЦПУ — тот самый, единственный чип для бюджетных ПК, думает **Алан Декстер...**

СПЕЦИФИКАЦИИ

- Разъем AM4
- Разрядность 64 бит
- Техпроцесс 14 нм
- Ядра 4
- Поток 4
- Тактовая частота 3,5 ГГц (3,7 ГГц турбо)
- Кэш L1 384 кБ, L2 2 МБ, L3 4 МБ
- Память DDR4, два канала, поддержка ECC
- PCIe 3.0 8+4 линий
- GPU Radeon Vega 8
- Тактовая частота GPU 1,1 МГц
- Тепловыделение 65 Вт

Сборщики бюджетных компьютеров еще никогда не были так избалованы. После многих лет жалкого прироста производительности и ужасной интегрированной графики мы обнаруживаем, что занимаемся чипами, которые год назад были бы высококлассными. Мы оперируем четырехъядерными процессорами с приличными частотами и интегрированной графикой.

Ryzen 3 2200G, как и 2400G (см. **LXF236**), использует ту же архитектуру, что и на дискретных графических картах AMD для обработки чипом 3D-графики. Если Vega 64 высшего класса имеет 64 вычислительных блока, а 2400G — 11, то здесь мы имеем дело только с восемью. Это соответствует базовой конфигурации 512 унифицированных шейдеров, 32 блоков отображения текстур и 16 выводов рендеринга.

В реальности это означает, что вы получаете от 30 до 60 кадров в секунду во вполне современных играх с разрешением 1080p и адекватными настройками игры. Интегрированная графика, предлагаемая здесь, не собирается бросать вызов серьезным дискретным предложениям, но достаточно мощна, чтобы вы могли играть в современные игры за небольшую часть цены. Это компромисс, на который готовы пойти многие.

Время турбо!

По части голой обработки, Ryzen 3 2200G удивляет мощностью, с учетом его низкой стоимости. За \$100 у вас есть четырехъядерный процессор, работающий на приятных 3,5 ГГц с максимальным разгоном 3,7 ГГц. И он преспокойно проработает на максимальном разгоне довольно долго. Стоит отметить, что, поскольку это процессор именно Ryzen 3, многопоточность не используется, так что эти четыре ядра соответствуют четырем потокам; но всё же это неплохо для бюджетного чипа.

Ryzen 3 2200G заменяет Ryzen 3 1200, предшествующий ему, втискивая графическое ядро в корпус AM4 и повышая рабочую частоту, а цену сохраняя при этом почти такой же (сейчас 1200 идет за £80). Частично причиной повышения рабочей частоты является то, что чипы первого поколения используют два основных комплекта, каждый из которых имеет пару рабочих ядер, а 2200G использует только одноядерный комплект в полной четырехъядерной сборке. Недостатком здесь является то, что кэша L3 не так много — 4 МБ вместо 8 МБ, хотя латентность и улучшена, и в повседневном использовании вы можете не ощутить это уменьшение.

Поскольку Ryzen 3 1200 уходит в отставку, любому начинающему бюджетному сборщику остаются на выбор два чипа: 2200G и 1300X, причем последний стоит £130. Беглое сравнение производительности показывает, что 2200G имеет преимущество над своим более дорогим собратом во всех тестах, которые мы запускали, кроме одного, но этот конкретный тест важен, поскольку подчеркивает недостаток нового чипа: линии PCI



Express снижены с 16x до 8x, чтобы освободить место интегрированной графике. Однако это снижение пропускной способности не должно быть проблемой, кроме как на очень высоком уровне.

По производительности 2200G проигрывает Intel Core i3-8100 в скорости процессора, но имеет преимущество, когда речь заходит об интегрированной графической производительности. А значит, если вы хотите играть в разные игры, но не хотите тратить серьезные деньги на отдельную видеокарту, этот новый бюджетный чип AMD — для вас.

Если игры вас не интересуют, или вы решили непременно иметь отдельную графическую карту, то чип Intel имеет преимущество над Ryzen. Здорово снова иметь настоящий выбор. Мы столкнулись с проблемами драйверов в Ubuntu 18.04 Beta (ядро 4.15) и Fedora 28 Beta (ядро 4.16), так как установка требует режима отката программного обеспечения. Fedora с Wayland имеет серьезные проблемы с производительностью, а Ubuntu после запуска работает хорошо, но падает на *Dirt Rally* при включении AA. К счастью, обновления ядра уже выпущены. **LXF**

ВЕРДИКТ

ПРОИЗВОДИТЕЛЬ: AMD
САЙТ: www.amd.com/ryzen
ЦЕНА: £85

ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	7/10
УДОБСТВО В РАБОТЕ	9/10
ОПРАВДАНОСТЬ ЦЕНЫ	10/10

Правильная цена, мощный интегрированный графический процессор и достаточно быстрый ЦПУ. Единственная слабость — недостаток линий PCIe.

» **Рейтинг 9/10**

Before the Storm

Толстовка с капюшоном, прыщавое лицо и бесцельное шатание по местному торговому центру; **Филлипа Уорр** отлично вписывается в команду **LXF**.

СПЕЦИФИКАЦИИ

Минимум

- ОС Ubuntu 16.04 64-bit, Steam OS
- Процессор Intel Core i3-2100, AMD Phenom X4 945
- ОЗУ 4 ГБ
- GPU AMD Radeon R7 250, Nvidia GeForce GTX 650
- На диске 14 ГБ

Рекомендовано

- Процессор Intel Core i3-6100, AMD Athlon X4 845
- ОЗУ 8 ГБ
- GPU AMD Radeon RX 460, Nvidia GeForce 1050

У этой подростковой драмы есть проблема, характерная для приквелов: обеспечить создание истории, которая прекрасно обошлась бы и без нее. *Before the Storm* [Перед бурей] берет персонажей из игры *Life is Strange* и погружает в их жизнь несколькими годами ранее. В этот раз основное внимание уделяется Хлое Прайс [Chloe Price] — неуклюжему комочку неразрешенных проблем, источнику подростковой ярости и постоянной прогульщице. Мы встречаем Хлою после смерти ее отца Уильяма, в момент, когда в ее жизнь врывается принцесса класса Рейчел Амбер [Rachel Amber].

Опыт в *Before the Storm* лучше оптимизирован, чем у предшественника. Основное внимание уделяется решению головоломок и излагается во всех подробностях отношение, находящееся в корне всего взаимодействия в *Life is Strange*.

Подростковые истины

Игру-оригинал любили, в частности, за то, что она, казалось, охватывала истины подростковой фантазии. Другими словами, она может быть чрезмерно драматизированной, эгоистичной или поглощенной осуществлением тайного желания. Встречаются места, где люди находят себя; осознают свою самоидентификацию; затевают конфликты, неосуществимые в реальности; вызывают в воображении выходы из фрустрации подросткового периода; становятся героями в собственных глазах.

Некоторые сцены сделаны на отлично. Хлоя может совершить вылазку в *D&D*, рассмешив нас до слез; есть участок импровизации с наблюдением за прохожими; и очень много кратких мигнов искренности, где усилия персонажей найти контакт друг с другом ощущаются очень достоверными.

Еще здесь много тайных желаний. В то время как *Life is Strange* развивает свое повествование вокруг центральной загадки, *Before the Storm* полностью сосредотачивается на истории Хлои и Рейчел. Первая игра поддерживала убедительные двусмысленности по части отношений. В этот раз мы напролом пошли в романтику. Это было здорово. Драматично и искренне и в изобилии насы-



Не каждый захочет вновь пережить моменты своих подростковых всплеск.

Иногда *Before the Storm* впадает в условную мелодраму с театрализованными злодеями (собственно, как и игра-предшественник). Что усиливает раздражение, поскольку всё играется в атмосфере #MeToo.

Откровенно отрицательные эмоции вызывают движения персонажей — особенно анимация бега или ходьбы, удивляющие странностью. На самом деле, у персонажей очень малое сходство с физическим миром.

Сопутствует этому дубовый язык диалогов. Между строками бывают неестественно длинные паузы, и налицо склонность говорить, а не показывать. Мы обнаружили, что во время некоторых сцен наши мысли где-то витали, тогда как сами обсуждения принимали заторможенный характер.

Разнообразны намеки и отсылки к исходной игре. Одни добавляют контекст к истории и персонажам; другие, наоборот, похожи на завлечение поиграть в другую игру. Есть также точки, где согласованность или логика уступают место эмоциям или эстетике.

Преимущество *Before the Storm* в том, что она более сфокусированная и более характерная пьеса, нежели *Life is Strange*. Она избавлена от большинства неуклюжих загадок, дарит приятные эмоции фанатам Хлои и достойным образом представляет девочку-подростка. **LXF**

Что? Моя толстовка была в стирке, ясно? Забей.



щенно знакомыми моментами. Предыдущий механизм перемотки времени был заменен резкими перепалками. Мы нашли это большим упущением, потому что вы по сути аргументируете свою точку зрения в диалоге неестественным образом.

ВЕРДИКТ

РАЗРАБОТЧИК: Feral Interactive

САЙТ: www.feralinteractive.com

ЦЕНА: £35

ИГРОВОЙ ПРОЦЕСС	7/10
ГРАФИКА	7/10
ПРОДОЛЖИТЕЛЬНОСТЬ	7/10
ОПРАВДАННОСТЬ ЦЕНЫ	7/10

Оглядка на события, лежащие в основе *Life is Strange*. По эмоциям сойдет, но иногда неуклюже.

» **Рейтинг 7/10**

softline®

Cloud Software Hardware Services

20+

Years in IT

IT-архитектура вашего бизнеса



корпоративные
мобильные решения
на платформе Microsoft

SaaS

CLOUD

PROJECTS

гибридные облака

SOLUTIONS

remote workplace

SAM

поставка аппаратных
решений

юридическая
поддержка

education

Mirapolis

SECURITY
учебный
центр

разработка IT-стратегий

hardware

ERP&CRM

CAD
CAM
CAE

VIRTUALIZATION

web-разработка legal support

**ЗАЩИТА
персональных данных**

SOFTWARE

публичные облака

САПР и ГИС

LINUX

**программное
обеспечение**

BI & BigData

SERVICES

WinSpace

**информационная
безопасность**

решения Microsoft

data leak
prevention

PaaS

разработка информационных систем

**корпоративный портал
DeskWork**

open
source

техническая поддержка

search

IaaS

частные
облака

protection

hosting

CONSULTING

technical support

HARDWARE

16+

8 (800) 232-00-23

+7 (495) 232-00-23

www.softline.ru

info@softline.ru

Мобильные новости

НОВИНКИ ARM

Мобильникам и ноутбукам

Процессор ARM Cortex-A76 на треть ускорит смартфоны и вдвое — ноутбуки с Windows.

Компания ARM утверждает, что представленный ею новый чип повысит производительность смартфонов на 35%, а производительность в задачах машинного обучения — вчетверо, при росте энергоэффективности на 40% (по сравнению с предыдущим чипом компании, Cortex-A75). Cortex-A76 выполнен на той же архитектуре v8.2, что и его предшественник, но имеет совершенно новую микроархитектуру, рассчитанную на экстремальную производительность и энергоэффективность. Это второй, после A75, чип, ядро которого поддерживает технологию DynamIQ big.LITTLE, позволяющую разработчикам однокристальных систем создавать процессоры из нескольких кластеров с различными процессорными ядрами. Cortex-A76 выполнен по техпроцессу 7 нм, его тактовая частота — 3 ГГц.

По части ноутбуков, ARM заявила, что модели на Cortex-A76 удвоят производительность по сравнению с текущими версиями, и получат гораздо более быстрый и плавный интерфейс пользователя для важных приложений, таких как пакет Microsoft Office.

ARM также представила GPU Mali-G76, для совместной работы с CPU Cortex-A76, способный содержать до 20 ядер. Относительно предыдущего Mali-G72, заявлен прирост производительности к площади GPU на 30–65%. Энергоэффективность выросла на 30%, скорость при машинном обучении — в 2,7 раза. GPU Mali-G76 поддерживает OpenGL ES до 3.2, Vulkan 1.1 и OpenCL до 2.0 Full Profile.



Cortex-A76 предназначен в первую очередь для мобильных устройств, но укомплектованные им ноутбуки будут не хуже устройств с процессорами Intel или AMD.

ВЗАИМОВЫГОДНО

Новая архитектура сети

«ВымпелКом» и Huawei заключили соглашение о развитии 5G.

ПАО «ВымпелКом» и компания Huawei подписали соглашение в области пилотирования и интеграции технологических решений 4,5G/5G и Интернета вещей. Стратегическое партнерство рассчитано на 2 года. «ВымпелКом» и Huawei намерены разрабатывать и тестировать технологии 4,5G и 5G для развития телекоммуникационной инфраструктуры пятого поколения. Партнеры протестируют решение Huawei Massive MIMO на коммерческой сети, позволяющей достигать гигабитных скоростей передачи данных в сетях LTE. Предусмотрен пилотный запуск технологии LTE License Assisted Access (LAA); протестируют технологии объединения несущих (Carrier Aggregation) между

сетями стандартов LTE TDD (Time Division Duplex) и FDD (Frequency Division Duplex). Будет проведен пилотно-коммерческий запуск технологии динамического переиспользования спектра (Cloud AIR), включая запуск LTE в диапазонах 900 и 2100, и разработаны решения Cloud RAN по функциям «тесного взаимодействия» базовых станций. Это повысит эффективность использования частотного ресурса «ВымпелКом» и улучшит опыт абонентов в местах с высокой концентрацией трафика. Также в 2018–2019 гг. пройдут пилотное тестирование технологии Nb-IoT, проработка возможностей сети поколения 5G с интерфейсом «New Radio» и тестирование на ней новых продуктов и услуг для абонентов «ВымпелКом».

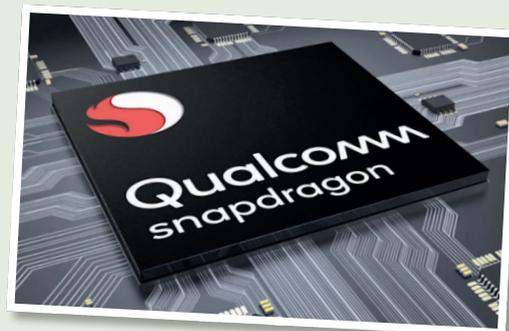
МОБИЛЬНЫЕ ПРОЦЕССОРЫ

Интеллект — в массы!

Qualcomm Snapdragon 710 добавляет самый современный ИИ в мобильные устройства среднего уровня.

SоC Qualcomm Snapdragon 710 — первый чип в новой 700-й серии, ориентированной на искусственный интеллект (ИИ). Согласно описанию, сделанному Qualcomm на Mobile World Congress в этом году, эти чипы будут чуть ниже 800-й серии по производительности, но обеспечат мобильные устройства среднего уровня самыми современными функциями ИИ. Snapdragon 710, как и Snapdragon 845, выполнен по технологическому процессу 10 нм. Это 8-ядерный чип, состоящий из двух мощных ядер Kryo 360 архитектуры Cortex-A75 с TЧ 2,2 ГГц и шести экономичных ядер Cortex-A55 с TЧ 1,7 ГГц. Qualcomm утверждает, что новый чип на 30% ускорится по сравнению со Snapdragon 660, с одновременным улучшением энергоэффективности на 40%; производительность ИИ в 710-м вдвое превосходит 660-й. Улучшенный фирменный процессор Hexagon Vector, подсистема визуальной обработки Adreno Visual Processing и CPU Kryo обеспечат оптимальный захват «контекстно-зависимых» фотографий и видео, выполнят «персонализацию голосовых и речевых шаблонов для более натурального взаимодействия».

Новый Qualcomm ISP Spectre 250 позволит получить фотографии «профессионального уровня» за счет поддержки 32-МП однообъективной и 20-МП двухкамерной установок. Система AI Engine поддерживает режимы “deer portrait” и Face Unlock с активным определением необходимой ГРИП. Поддерживается воспроизведение 4K HDR — впервые вне топовой 800-й серии, технология Qualcomm Quick Charge 4.0+ и скорость загрузки до 800 Мбит/с благодаря новому модему Qualcomm Snapdragon X15 LTE. Согласно Qualcomm, «многие OEM-производители» планируют выпуск устройств на базе Snapdragon 710 уже во II квартале этого года. Ранее чипмейкер заявлял, что разрабатывает серию Snapdragon 700 для удовлетворения «растущих потребностей экосистемы смартфонов Китая», и вполне резонно предположить, что одними из первых применят новый чип Huawei и Xiaomi.



Мобильная платформа Snapdragon 710, первая в новой и очень значительной 700-й серии, делает доступными технологии и функции, ранее реализованные только на устройствах премиум-класса.

В МИРЕ WEAR OS

Который час может быть сейчас?..

Mobvoi показала «умные» часы класса luxury.

Bнастоящее время выпуск «умных» часов Mobvoi Pro запланирован только для рынков Китая. Секрет новой версии заключается во втором прозрачном дисплее FSTN-LCD, расположенном над основным «циферблатом» и отличающемся крайне низким энергопотреблением — гипотетически, если пользоваться только им, то часам не потребуется подзарядка в течение нескольких дней. Прочий функционал включает NFC, фирменный встроенный GPS и другие привычные пользователям Wear OS «плюшки». Устройство активно решает, какой дисплей следует использовать в той или иной ситуации, «бесшовно» переключая LCD и OLED дисплеи при необходимости. Безусловно — интересное решение, учитывая, что основная (и серьезная!) проблема устройств под управлением Wear OS (бывшая Android Wear) — необходимость ежедневной (а иногда и более частой) зарядки. Mobvoi уже подтвердила, что цена новинки будет около \$300 (по этому показателю Mobvoi Pro поставлены наравне с Huawei Watch 2), а LTE-версия может появиться позднее.

Ранее Mobvoi представила в Великобритании свои «умные» часы Ticwatch моделей E и S. Обладая выразительным тонким дизайном и широким диапазоном цветовых решений, эти часы предлагают большинство привычных и ожидаемых пользователем от подобных устройств функций, за исключением NFC (справедливости ради, следует отметить, что платежный сервис Google пока полноценно работает на платформе Wear OS только в нескольких странах мира, в т.ч. США, Великобритании, Канаде, Австралии и Испании). С 18 мая к этим странам добавились Польша и Россия). Ценовая политика Mobvoi на рынке UK заключается в том, что Ticwatch продаются по ценам, значительно меньшим, чем аналогичные модели конкурентов (начальная цена менее £150). Функциональность моделей E и S практически идентична, разница в цене связана, в основном, с улучшенной эстетикой версии S.



Компания Mobvoi надеется, что ее новые smart-часы смогут изменить сложившуюся вокруг Wear OS печальную ситуацию...

В настоящее время в самом разгаре находится краудфандинговая кампания Mobvoi по сбору средств на создание TicPods Free — эквивалента AirPods на Android. **LXF**

Сравнение

CMS Made Simple » Concrete5 » Drupal »
Exponent CMS » Joomla



Шашанк Шарма

днем выступает в суде в Нью-Дели, а по ночам вершит суд над открытым кодом!

Управление контентом

Готовы урвать себе местечко онлайн, но пока не выбрали систему управления контентом? **Шашанк Шарма** может вам кое-что предложить.

ПРО НАШ ТЕСТ...

Для CMS требуется только должным образом настроенный web-сервер и база данных. У вас есть большой выбор в зависимости от типа сайта, который вы хотите создать. Для *Сравнения* этого месяца нас интересовали только те CMS, которые помогают создавать сайт целиком, а не специализированные системы для блогов или wiki.

Выбор базы данных, web-сервера и языка скриптов не особо важен для конечного пользователя, но мы для нашего *Сравнения* выбрали только CMS, основанные на PHP. Помимо web-сервера *Apache*, мы использовали для нашей базы данных *MySQL*.

Мы будем тестировать проекты по количеству предлагаемых ими функций по умолчанию, простоте установки и по наличию поддержки дополнений для расширения функциональности. С учетом постоянно растущего числа людей и предприятий, выходящих онлайн, необходима также приличная документация, чтобы помочь новичкам внедрить и настроить свою CMS. Не менее важны и опции настройки, предлагаемые каждым сервисом.



Вот уже не один год создание блога сводится практически всего лишь к созданию учетки на одном из бесплатных сервисов. Но если вы решите создать сайт посложнее, с богатым контентом, вам придется делать его самому. Это тоже несложный процесс — если вы не против потратить некоторое время на создание должной среды. Но когда вы выберете web-сервер и создадите базу данных, вам придется решить, какую систему управления контентом (CMS) взять для своего сайта.

В зависимости от выбранного вами языка скриптов вы можете предпочесть CMS, построенную на Ru-

by, Python, Java и прочих; однако самым популярным выбором остается PHP. Как нам ни хотелось представить разные языки, самонавязанное ограничение до полнофункциональных CMS, недавно выпустивших новый релиз, не оставило нам выбора.

Поскольку всё больше пользователей подключаются к Сети через разнообразные устройства самых разных форм-факторов, ваш сайт должен хорошо работать на всех этих устройствах. Современные CMS нередко даже оптимизированы для мобильных устройств. Идеальная CMS легко разворачивается и позволяет бесконечно настраивать ее, чтобы получить в итоге результат по своему желанию.

Простота развертывания

Сможете ли вы установить их сами?

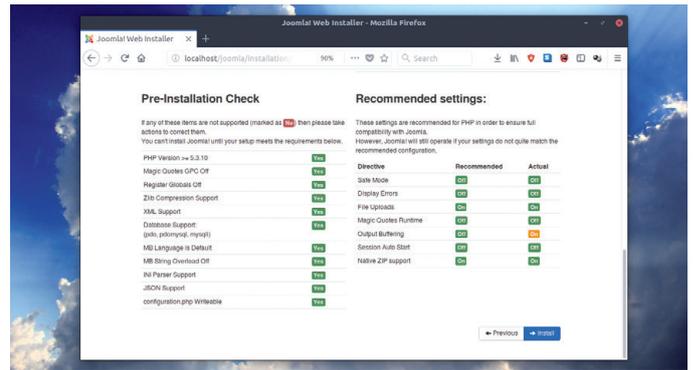
С постоянно растущим числом ответвлений и альтернатив у пользователей имеется выбор из нескольких web-серверов и программ DBMS. Большинство CMS вполне устроят поддерживаемые экземпляры web-сервера *Apache*, база данных *MySQL* и *PHP*.

Хотя вам не придется в совершенстве овладеть любым из них, чтобы установить и использовать CMS, вы должны быть знакомы с ключевыми концепциями, чтобы подготовить среду для установки. В дополнение к программе, большинство CMS также требуют установки определенных библиотек и их включения в *PHP*.

Как минимум, вы должны создать базу данных и пользователя для управления этой базой данных, потому что вас попросят указать эту информацию во время установки. У всех CMS в нашем списке имеется сетевой мастер установки, который проведет вас по разным шагам настройки вашего сайта.

Как понятно по названию, *CMS Made Simple* невероятно проста в настройке. Самое лучшее в проекте — его подробная документация, которая поможет настроить сайт даже новичку, никогда не работавшему с базой данных или web-сервером.

Exponent CMS требует специальной настройки *MySQL*, без которой установка не удастся. К сожалению, документация не предлагает обходного пути или хотя бы упоминает эту потенциальную проблему. Если вы выберете *Exponent CMS*, перед началом установки обязательно посмотрите, как отключить режим *Strict* в *MySQL*. Хотя установщик советует вам поднастроить *MySQL*, чтобы продолжать установку в случае отказа, внести изменения и продолжить установку возможности нет. Единственная опция — начать установку



Все проекты запускают предустановочную проверку, которая сообщит вам, нужны ли дополнительные библиотеки или какая-либо настройка.

Еще хуже то, что проект не позволит вам повторно использовать базы данных, которую вы для него уже создали, и вам придется либо опустошить базу данных, либо создать новую.

В дополнение к сетевому мастеру установки *Concrete5*, *Drupal* и *Joomla* можно также установить с помощью *Composer*, менеджера зависимостей для *PHP*. Если вы не работали с ним раньше, простейшей опцией будет выбрать сетевой мастер установки. Причина в том, что использование *Composer* для установки этих проектов не слишком хорошо документировано, и если у вас возникнут проблемы, вам придется искать решение в Сети.

ВЕРДИКТ

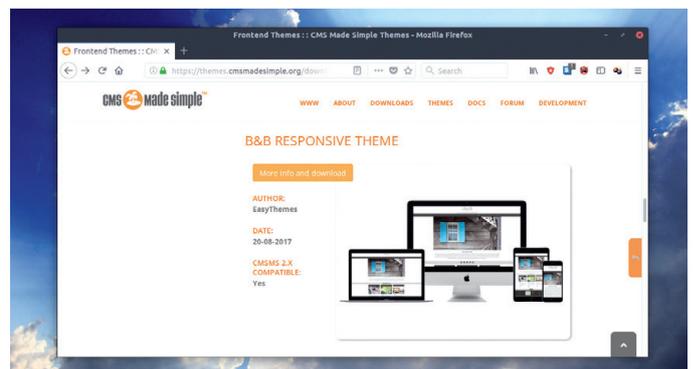
CMS MADE SIMPLE	10/10	EXPONENT CMS	7/10
CONCRETE5	9/10	JOOMLA	9/10
DRUPAL	9/10		

Кроме CMSMS, остальные проекты не предлагают подробностей установки.

Поддержка устройств

Доступ к массам.

В своей вступительной речи на конференции *Drupalcon Chicago* в 2011 г. Дриес Бейтерт [Dries Buytaert], основатель *Drupal*, наилучшим образом подытожил важность мобильных устройств для CMS, заметив, что если бы ему пришлось «создать *Drupal* с нуля сегодня, я бы в первую очередь озаботился работой с мобильными устройствами, а уж во вторую — с настольным ПК». И это еще более значимо сегодня, когда несколько последних исследований показали, что более 60% пользователей используют для доступа к сайтам свои мобильные устройства, а не настольные ПК и ноутбуки. Адаптивный web-дизайн обеспечивает хороший рендеринг сайтов на разных устройствах. С апреля 2015 г. Google выбраковывает недружелюбные к мобильным устройствам сайты и страницы. К счастью, все CMS в нашем списке позволяют создавать адаптивные сайты. Т.е. они поддерживают разные элементы, например, адаптивные темы и изображения. Последнее очень важно, иначе изображения не будут автоматически масштабироваться под размер экрана устройств. Проект *CMS Made Simple* предлагает несколько готовых для мобильных устройств тем и настроек. Члены сообщества также делятся собственными адаптивными темами и настройками. Подобным же образом *Exponent CMS* предлагает ряд тем, разработанных для мобильных устройств. Этот проект предлагает *Twitter Bootstrap Theme*, а также несколько модулей, пригодных для подготовки вашего мобильного сайта.



Есть несколько адаптивных тем для *CMS Made Simple*, *Concrete5* и других проектов.

Все остальные представленные здесь проекты осознают важность мобильных устройств. Помимо бесплатных модулей и тем, *Joomla* и *Concrete5* также предлагают несколько коммерческих дополнений, которые помогут вам настроить свой сайт для комфортного общения с мобильными платформами.

ВЕРДИКТ

CMS MADE SIMPLE	10/10	EXPONENT CMS	10/10
CONCRETE5	10/10	JOOMLA	10/10
DRUPAL	10/10		

Дружелюбие к мобильным устройствам — достоинство всех наших проектов.

Взаимодействие с пользователем

Просты ли они в работе и управлении?

С MS созданы ради выполнения одной функции: чтобы помочь вам создать сайт без особых проблем. После установки CMS не должна мешать вам в создании контента. Но CMS — сложная программа. Огромное количество элементов и настраиваемых опций для обычного современного сайта делают администрирование сложной задачей. Хорошая CMS должна найти верный баланс между созданием удобства и для пользователя и для администратора.

От создания пользователей и управления их правами до расширения функциональности и выполнения таких задач, как создание резервных копий и обновлений, интерфейс администрирования должен суметь помочь вам логичным способом осуществлять навигацию между элементами.

С помощью боковой панели или мозаики в Dashboard или через выпадающие меню, каждая CMS в нашем списке нашла способ не напугать пользователя своими возможностями администрирования.

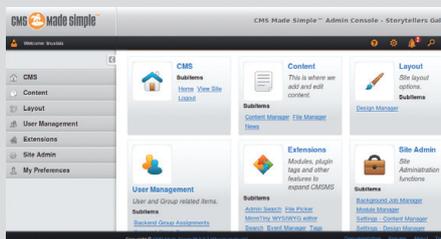
CMS Made Simple

10/10

Интерфейс администратора предлагает сжимаемую боковую панель для быстрого доступа к разным разделам, таким как User Management [Управление пользователями] и Layout [Структура]. Нажмите Site Admin > Settings - Global Settings [Администрирование сайта > Настройки - Общие настройки], чтобы настроить почту и прочие важные пункты, например, включить режим техобслуживания для сайта. Верхняя панель также предлагает быструю ссылку на официальную документацию и то, что проект именуется необработанными уведомлениями [unhandled notifications]. Проект настаивает на удалении файлов установки и публикует уведомление, если вы этого не делаете.

Чтобы вернуться на домашнюю страницу Dashboard, нажмите Home в CMS в левой боковой панели. Вы можете создать новый контент с помощью менеджера контента, который предлагает предпросмотр, так что вы сможете увидеть, как он будет выглядеть, и при необходимости что-то исправить.

Если вы никогда прежде не использовали CMS, с CMSMS вы будете чувствовать себя как дома.



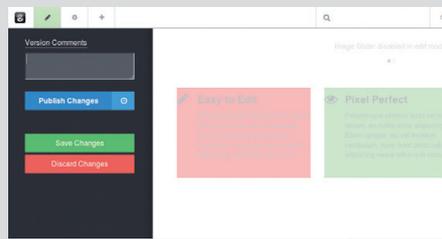
Concrete5

10/10

Интерфейс администрирования Concrete5 предлагает боковую панель справа. На главном экране имеются слайды популярного инструмента, предложения дня на рынке и имеющиеся дополнения и тема.

Когда вы войдете в систему как администратор, при посещении вами вашего сайта панель вверху предлагает кнопки в помощь редактированию страницы или для добавления нового контента/страниц. Проект поддерживает непосредственное редактирование контента прямо на месте [inline]. Вам не придется перемещаться по интерфейсу администрирования, чтобы отредактировать контент вашего сайта: это можно сделать, нажав на элемент, например, боковую панель, и на Edit [Редактировать]. Вы также можете внести изменения в дизайн каждого элемента, например, цвет фона, нажав на элемент и затем нажав Design [Дизайн].

Проект отслеживает все изменения, внесенные вами в определенную страницу, и при желании вы сможете вернуться к предыдущей версии страницы.



Документация и поддержка

Пользователи не могут RTFM, если его нет!

Помимо официальной документации и форумов, представленные в данном Сравнении проекты настолько популярны, что вы легко найдете составленные пользователями руководства и справочники по разным темам. Более старые проекты, такие как Drupal и Joomla, также являются темой для парочки книг.

CMS Made Simple сильнее всех впечатлила нас своей документацией, по всем аспектам настройки системы к установке. Официально предлагаются пошаговые инструкции по установке и настройке и другим важным темам: обновление, управление пользователями, навигация по структуре и т. д. На форумах кипит активность, и помимо общих дискуссий и обсуждений конкретных модулей вы найдете советы и хитрости от членов сообщества.

В документации Exponent CMS упоминаются основные шаги перед установкой, например, создание базы данных и пользователя, но нет инструкций, как это сделать, для новичков, незнакомых с подобными технологиями. Правда, на сайте есть ссылка на явно устаревшую wiki, но и та недоступна.

Как и у Exponent CMS, документация Drupal и Joomla почти целиком сосредоточена на том, как использовать эти проекты по максимуму.

Помимо подсказок пользователям по созданию базы данных и настройке сервера, нет никаких пошаговых инструкций в помощь менее опытным пользователям.

Если судить по их документации, эти проекты, кажется, совершенно не заинтересованы в привлечении новых пользователей, поскольку в них нет никакого введения в CMS и сопутствующие технологии.

Concrete5 предлагает очень дельную документацию по ряду тем. Кроме того, здесь имеется множество руководств для редакторов, разработчиков и дизайнеров.

ВЕРДИКТ

CMS MADE SIMPLE	9/10	EXPONENT CMS	7/10
CONCRETE5	7/10	JOOMLA	7/10
DRUPAL	7/10		

Форумы, руководства и видео на YouTube – не повод презирать нужды новичков.

Drupal

10/10

Exponent CMS

9/10

Joomla

10/10

Панель администрирования сверху страницы предлагает несколько вкладок: Content [Контент], Structure [Структура] и Appearance [Внешний вид]. В отличие от остальных проектов, *Drupal* создает локальный репозиторий своей документации и предлагает ссылку на него в верхней панели.

Нажмите Configuration [Настройка], чтобы настроить задания *cron* по выполнению обслуживания, переведите сайт в режим поддержки, импортируйте/экспортируйте настройку, редактируйте сайт на предмет производительности, включая кэширование, и т.д. Вы можете быстро добавлять новый контент к своему сайту, нажав на вкладку Content. Проект также создает ряд важных отчетов, например, о проблемах безопасности, доступ к которым можно получить, нажав на Reports > Status Reports [Отчеты > Отчеты о состоянии]

Вас проинформируют обо всех обновлениях для основных программ или для любого установленного модуля, когда вы нажмете Configuration. Вы также можете запланировать для проекта задание *cron*, чтобы автоматически проверять наличие обновлений.

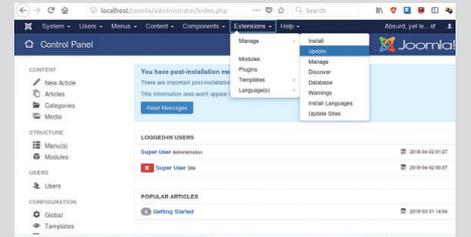
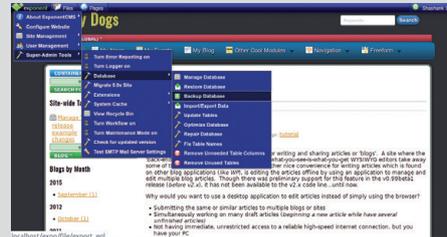
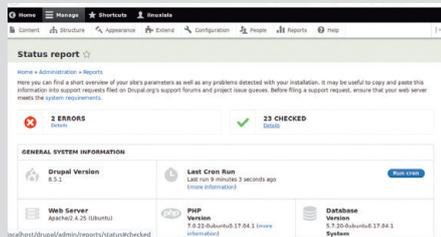
Вы получаете информацию об имеющихся заплатках для основных программ, как только входите в систему как администратор.

В отличие от других CMS, предлагающих специальный интерфейс администратора, доступ к функциям администрирования *Exponent* можно получить, нажав на кнопку Exponent. Она откроет выпадающее меню, предлагающее доступ к User Management, Site Management, и т.д. Super-Admin Tools — универсальное меню для всех остальных функций, таких как управление расширениями, обновление базы данных и переключение в режим поддержки сайта.

Как и *Concrete5*, *Exponent* поддерживает непосредственное создание и редактирование контента. Возможно, на привыкание к нему уйдет некоторое время, поскольку вместо кнопок проект часто использует текстовые ссылки, например, Add a New Blog Article [Добавить новую статью блога] или Edit [Редактировать]. Если ваша страница богата контентом, очень легко упустить эти ссылки, разве что вы будете заранее знать, что хотите найти.

Даже если у пользователя есть административные привилегии, экран приглашения на главном сайте не выдаст доступа к интерфейсу администратора. Надо нажать на Site Administration [Администрирование сайта], после чего вас попросят снова войти в систему. В качестве альтернативы можете ввести в свой браузер <http://127.0.0.1/site/administrator>. Интерфейс администратора *Joomla* называется Control Panel [Панель управления] и предлагает ряд выпадающих меню на верхней и боковой панелях. Обновления основных программ и установленных расширений перечислены в разделе Maintenance [Обслуживание] на боковой панели.

Вложенные меню на верхней панели дают более быстрый способ навигации по панели управления. Так, щелчок по Users [Пользователи] на боковой панели переместит вас на страницу пользователей, где для создания нового пользователя надо нажать New [Новый]. Сравните это с верхней панелью, где переход на Users > Manage > Add new user [Пользователи > Управление > Добавить нового пользователя] переместит вас на ту же страницу.



Настройка и темы

Разнообразие придает жизни вкус.

В отличие от других CMS, которые позволяют вам войти в учетку администратора подобно любому другому пользователю, для *CMS Made Simple* и *Exponent CMS* вы должны вручную направить браузер на <http://127.0.0.1/site/admin> или <http://127.0.0.1/site/login.php> соответственно, если хотите зайти на сайт.

За исключением *Drupal*, все CMS во время установки предлагают образец контента, и эта опция облегчает вам доступ к разнообразным темам по умолчанию.

В дополнение к девяти разным дизайнам, предлагаемым по умолчанию, с *CMS Made Simple* вы можете даже загрузить другие темы, предложенные пользователями, и некоторые из них платные. Однако помните, что они не протестированы, и проект не дает никаких гарантий на эти сторонние предложения. Каждая тема сочетает в себе много разных элементов, именуемых шаблонами, например, колонтитул внизу страницы, навигационные цепочки, результаты поиска и т.д., которые вы можете перетащить в свой дизайн. Для более серьезной настройки, например, смены цвета фона и т.п., надо вручную редактировать соответствующие файлы.

Exponent CMS предлагает ряд тем по умолчанию, и документация предлагает инструкции по созданию индивидуальной темы. Однако нет возможности скачивать и переключать темы из интерфейса администратора.

Concrete5 сильно облегчает возможность включить отдельные темы для пользователей мобильных устройств и настольных ПК. Проект предлагает на выбор несколько бесплатных и коммерческих опций, и вы можете установить их одним нажатием, не покидая сайта.

Для настройки внешнего вида *Joomla* надо или самостоятельно создать свой дизайн, или использовать предложения от третьих сторон.

ВЕРДИКТ

CMS MADE SIMPLE	8/10	EXPONENT CMS	5/10
CONCRETE5	10/10	JOOMLA	6/10
DRUPAL	10/10		

Drupal и *Concrete5* предлагают наибольшее число опций настройки посредством тем и прочих расширений.

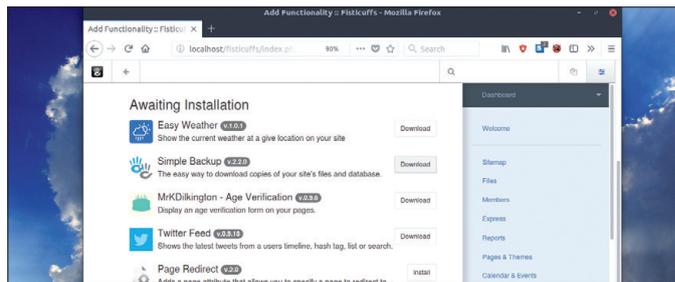
Расширения

Когда вы в настроении поглазеть на витрины...

Разыскивая модули для расширения функций сайта *Drupal*, вы можете фильтровать результаты по статусу поддержки или разработки проекта, по совместимости версий или по определенным категориям, таким как Content [Контент], E-commerce, Advertising [Реклама], Education [Образование], Media [Медиа], SEO и т.д. И очень хорошо, что они есть, потому что без всяких расширений простенький сайт *Drupal* вряд ли произведет на кого-то незабываемое впечатление, включая даже и самого автора сайта!

У *Joomla* тоже предусмотрен сайт, посвященный расширениям, которые подразделяются на разные категории, такие как Style & Design [Стиль и Дизайн], Site Management [Управление сайтом], Social Web [Соцсети] и Photos & Images [Фотографии и изображения].

Панель администратора на вашем сайте *Concrete5* предлагает ссылки и введение в имеющиеся дополнения. Вы должны создать учетную запись пользователя в магазине проекта [Marketplace]. Можно фильтровать дополнения по цене или ограничить результаты только бесплатными расширениями. У вас также будет выбор из нескольких категорий, как в других проектах. Однако, в отличие от других, *Concrete5* также позволяет определить уровень умений и ограничить результаты по уровням: Beginner [Начинающий], Intermediate [Средний], Expert [Эксперт] или Bleeding Edge [Супер Продвинутый]. Для каждого добавления магазин дает пользователям возможность загрузить обзор и выставить рейтинг. Обязательно прочитайте документацию для



Приобретая расширения в магазине *Concrete5*, вы должны скачать их с Dashboard > Extend Concrete5 > Add functionality page [Dashboard > Расширить Concrete5 > Добавить страницу функций].

навигации по Marketplace, или вы будете просто добавлять товары в корзину, но дальше этого не уйдете.

CMS Made Simple предлагает массу разных плагинов и модулей для расширения функций основных программ. К сожалению, большинство этих дополнений уже не первой свежести, и использовать их небезопасно. Документация предлагает быстрое знакомство с семью расширениями, являющимися частью установки по умолчанию, но на том и всё.

ВЕРДИКТ			
CMS MADE SIMPLE	5/10	EXPONENT CMS	10/10
CONCRETE5	10/10	JOOMLA	10/10
DRUPAL	10/10		

Репозиторий расширений *Exponent CMS* размещает массу тем и по большей части устаревшие дополнения.

Основные функции

Можно ли использовать их с ходу?

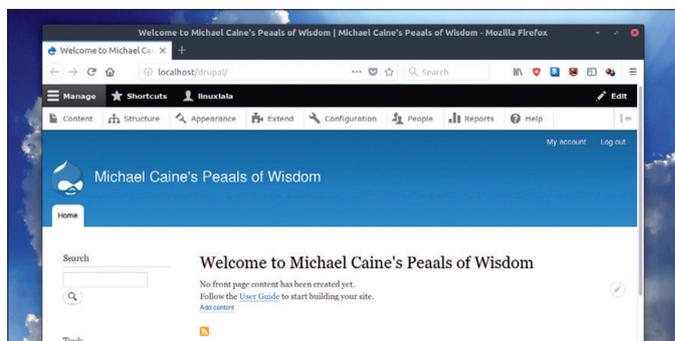
Все CMS предлагают расширенные опции управления пользователями, позволяющие определять права каждого зарегистрированного пользователя. Вы также можете найти дополнительные расширения контроля доступа для всех проектов, кроме *CMSMS* и *Exponent*. Проекты также имеют дополнительные функции безопасности, такие как блокировка указанных IP-адресов и создание черных списков.

Советуем потратить некоторое время на ознакомление с CMS, прежде чем решать, подходит ли она вам. Даже если CMS делает всё, что вам нужно, вас может не устроить ее интерфейс администратора, или наоборот.

Мы обращаем на это особое внимание, потому что *Exponent CMS*, хотя и очень хороша, предлагает очень специфический подход к дизайну сайта. Она определила все элементы обычного сайта как модули. Блоги, форумы, формы поиска, лента новостей соцсетей и т.д. — всё это определяется как модули. Проект также позволяет включить и разместить модули в разных разделах страницы, например, на верхней панели, наверху, в середине или внизу боковой панели и т.д.

С установкой *Drupal* по умолчанию вы вряд ли сможете сделать что-то большее, нежели создать статическую страницу. К счастью, проект по умолчанию устанавливает несколько важных расширений, но по умолчанию они отключены.

Поскольку *Joomla* не предлагает никаких важных расширений, установленных заранее, вы должны вручную прошерстить репозитории на предмет поиска дополнений, обеспечивающих нужные вам функции. Не следует считать, что простые предложения *Drupal* и *Joomla* — это обязательно плохо;



Вы можете считать *Drupal* и *Joomla* с их минималистским пост-установочным предложением Arch Linux в мире CMS. Обе отлично настраиваются.

нет, ведь они дают вам возможность построить сайт с самого основания по вашим собственным спецификациям.

Установка по умолчанию *Concrete5* поможет вам запустить блог или сайт с большим количеством медиа, и кроме того, предлагает ряд важных функций, а также разные опции оптимизации SEO, Captcha и прочие меры по предотвращению спама. Вы даже можете добавить список запрещенных слов и переклочить сайт в режим обслуживания.

Как бы мы ни ценили свободу создания собственного сайта с помощью *Drupal* и *Joomla*, нельзя переоценить внимание *Concrete5* к основным функциям безопасности и прочим настройкам.

ВЕРДИКТ			
CMS MADE SIMPLE	8/10	EXPONENT CMS	8/10
CONCRETE5	10/10	JOOMLA	8/10
DRUPAL	8/10		

Установки *CMSMS* по умолчанию хватит большинству пользователей, даже SMB.

CMS

Вердикт

Сраница Wikipedia, где перечислены Системы управления контентом с открытым кодом, называет десятки проектов, и выбрать из них пять объектов тестирования для *Сравнения* этого месяца было само по себе крайне непростой задачей. Мало того, что мы сами ограничились проектами, у которых релиз вышел не очень давно; мы также сузили свой охват CMS, способными удовлетворить потребности разных пользователей. Нам были нужны CMS для отдельных пользователей, но при этом удобных и для небольших предприятий, и даже школ.

Поскольку со временем сайты растут, нам также были нужны CMS, предлагающие средства для резервного копирования конфигурации и базы данных и даже их импорта, если возможно. Вероятно, важнее всего были доступность плагинов и простота развертывания.

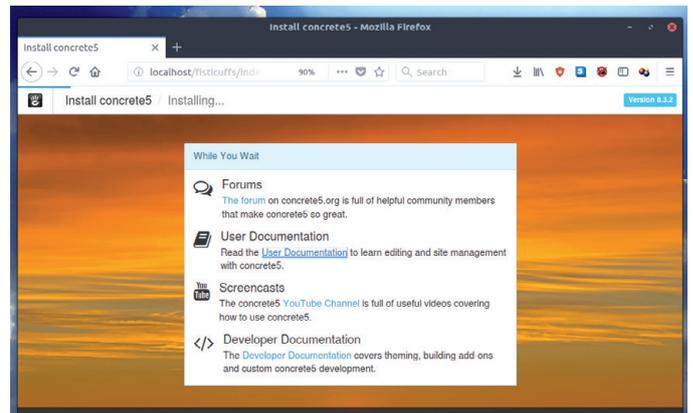
Поскольку скорость работы постоянно увеличивается, такие проекты как *Drupal*, *Joomla* и *Concrete5* предлагают онлайн-демо, чтобы вы могли с ними ознакомиться и принять информированное решение о том, какую из них выбрать для своего сайта.

Хотя никого нельзя наказывать за непохожесть, именно так мы поступили с *Exponent CMS*. Хотя проект надежный и простой в использовании, он сильно отличается от большинства других CMS, и некоторые пользователи могут счесть это неприятным. Слишком сжатая документация и ограниченные расширения также являются сильными негативными факторами.

Joomla и *Drupal* в общем не совершили ничего предосудительного. Отсутствие официальных тем для *Joomla* были достаточной причиной убрать ее с пьедестала. *Drupal* похож на Arch Linux своей прочностью и надежностью, однако его имеющаяся документация совершенно непригодна для начинающих пользователей, поэтому он занимает третье место.

CMS Made Simple — истинное удовольствие в использовании, и в равной мере подходит начинающим пользователям и опытным.

Однако внимание *Concrete5* к SEO, безопасности и его богатый рынок ПО в купе с линейным редактированием выделяет ее среди прочих CMS в этом тесте. Это, однако, не значит, что у проекта нет недостатков. Фактически, он так же повинен в игнорировании новичков, как *Exponent*, *Drupal* и *Joomla*. Кроме того, требование сначала купить нечто на рынке, а потом отдельно скачать это с собственного сайта, как минимум, странно. **LXF**



1^е место **Concrete5** **10/10**

Версия: 8.3.2 **Сайт:** www.concrete5.org **Лицензия:** MIT
Все CMS могут поучиться у *Concrete5*.

2^е место **CMS Made Simple** **9/10**

Версия: 2.28.4 **Сайт:** <https://cmsmadesimple.org> **Лицензия:** GPL
Нацелена на новичков, но отсутствие расширений разочаровывает.

3^е место **Drupal** **8/10**

Версия: 8.5.1 **Сайт:** www.drupal.org **Лицензия:** GPL
Чистый победитель, если вам нужна глубоко настраиваемая CMS.

4^е место **Joomla** **8/10**

Версия: 3.8.6 **Сайт:** www.joomla.org **Лицензия:** GPL
Впечатляет, но ей стоит поучиться трюкам *Concrete5*, чтобы не отстать.

5^е место **Exponent CMS** **7/10**

Версия: 2.4.2 **Сайт:** www.exponentcms.org **Лицензия:** GPL
Документация и плагины по-прежнему ее основная слабость.

» РАССМОТРИТЕ ТАКЖЕ

Есть куча специализированных CMS, которые мы не можем охватить в этом *Сравнении*. Одной из них, которая показалась нам весьма привлекательной, была *Wagtail*. Написанный на Python, этот проект использует среду Django и заявляет, что создан разработчиками для разработчиков. И этого было достаточно, чтобы не включить его в нашу подборку, а его документация, которая местами устарела, окончательно решила его судьбу. Мы также не хотели брать CMS, предназначенную исключительно для электронной коммерции, а если вам нужно именно это, стоит обратить

внимание на *WebGUI*, которая написана на Perl, но не видела нового релиза более двух лет.

Если вы хотите настроить CMS без необходимости сначала создавать базу данных, вы, возможно, захотите познакомиться с *CMSimple*, которая хранит контент в файлах HTML и базы данных не требует. За последние несколько лет проект вдохновил создание нескольких ответвлений, но и они, как и сама *CMSimple*, не видели новых релизов уже несколько лет.

UBUNTU 18.04 ОСВОИМ!

Итак, прибыл свежий Ubuntu, с совершенно новым рабочим столом вам на радость и удивление.

Джонни Бидвелл раскатывает оранжевый ковер...

Успокой на миг свои суетные мысли, дорогой читатель, и скажи, что ты слышишь. Ласточек, стрижей и каменок, которые возвращаются к своим северным гнездам? Нет, прислушайтесь внимательнее. Слышите механический грызущий звук вон от тех деревьев?

Это биомеханический бобер, и, может быть, несколько гномов, и это может значить только одно: вышел самый новый LTS-релиз Ubuntu. По традиции, наша обложка выдержана в оранжевом цвете, наш редактор диска целыми днями тестировал дневные сборки, а Джонни, не замолкая, болтал о Вальпургиевой ночи. Но кое-что изменилось с того момента, как два года назад вышел последний LTS.

Тогда все были взволнованы дерзкими проектами рабочего стола Canonical: Unity 8, Mir и конвергенция. Все они теперь переданы сообществу, а Ubuntu, после семи лет своего пути, вернулся к рабочему столу Gnome. Противоречивый ход, но и прагматичный. Canonical нацелился на IPO, а амбициозные проекты рабочего стола, не дающие прибыли, инвесторов не привлекают.

Более того, Gnome 3 стал зрелым и стабильным — и больше не является тем устрашающим чудовищем рабочего стола, которым был когда-то; а Canonical проделал отличную работу по его индивидуальной настройке, чтобы сделать его как можно удобнее для бывших пользователей Unity. Gnome также возглавляет движение по переходу на Wayland, преемник устаревшего

сервера отображения X.org, поэтому Ubuntu 18.04 — отличный дистрибутив для отважных сердец и любителей пробовать всё первыми, чтобы поэкспериментировать с будущим стекком отображения (не вздрагивайте, это пока еще не выбор по умолчанию).

Конечно, если Gnome вас всё же не устроит, хватает и других рабочих столов. Мы расскажем о самых интересных и, конечно, о самых интересных изменениях внутри — новом ядре, snap-пакетах, не говоря уж о сервере Ubuntu, облаке и предложениях IoT. И мы не забыли об упрямах, упорно работающих с 32-битным оборудованием. Может, 32-битного средства установки уже и нет, но это вовсе не означает, что вы не сможете пожать урожай нового Ubuntu!



Бодрые бобры

Вокруг последнего релиза Ubuntu под названием Bionic Beaver [Бионический Бобер] шумихи очень много. Что же там нового? Давайте выясним...

После 14 лет выходов релизов Ubuntu вы можете считать это событие рядовым. И ошибаетесь: это важная новость. Во-первых, это релиз LTS, и вполне вероятно, что он станет частью жизни многих на следующие пять лет. Более того, ряд других популярных дистрибутивов рабочего стола основан на Ubuntu LTS, так что данный релиз формирует и их базу. И, наконец, в этом релизе мы видим возвращение Ubuntu к рабочему столу Gnome после семи лет следования своим путем с Unity.

Предосторожности

Установка Ubuntu, как обычно, проста. Однако это новый релиз, так что не стоит ставить его на системы, от которых зависит ваш ежедневный труд. Перед установкой убедитесь, что всё работает в среде live. И сделайте резервные копии всех важных данных, прежде чем нажать на кнопку установки.

Также не используйте наш установочный диск. Скачайте обновленный ISO-образ с www.ubuntu.com/download/desktop. Этот совет может показаться странным, особенно при потраве кошелька на журнал, но внутри этих обложек мы обязаны, как минимум, проявить благоразумие. Из-за прискорбной необходимости печатать журнал, это не последний релиз, а дневной образ середины апреля. Мы протестировали его с наивозможной тщательностью, но не можем гарантировать отсутствия какой-нибудь невыявленной ошибки, которая прикончит ваш BIOS (как это было в 17.10) или другую операционную систему. Вы можете использовать утилиту *zsync* для обновления ISO с минимальной передачей данных

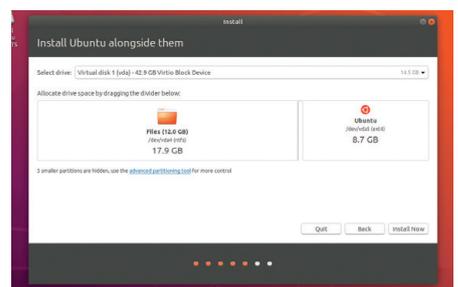
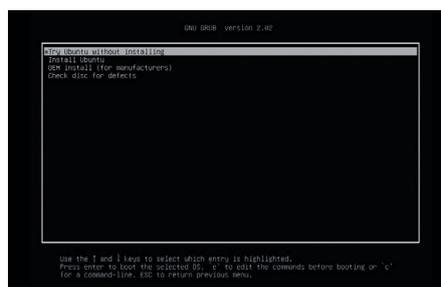


Программа установки Ubiquity работает, как обычно, ровно, и нам понравилось оформление – бобер, построенный из окружностей.

(см. внизу). Она скачает только измененные части, что должно составить всего несколько мегабайт. После завершения прожгите обновленный ISO на диск с помощью своей любимой утилиты или запишите на USB с помощью *dd* или *Etcher*.

Когда вы начнете установку, вам, возможно, порадует опция минимальной установки. Она не сэкономит вам особо много места (около 500 МБ), но уберет большую часть громоздких приложений, оставив только браузер и основные инструменты. Затем вы можете добавлять что хотите, и это просто отлично, если ваши симпатии к приложениям отличаются от предлагаемых Ubuntu по умолчанию.

УСТАНОВКА UBUNTU



1 Скачайте официальный ISO

Если у вас малая пропускная способность канала и уже стоит Linux, скопируйте ISO с нашего диска (в директории **Ubuntu/**), установите *zsync* и обновите ISO:

```
$ chmod 755 bionic-desktop-amd64.iso
$ mv bionic-desktop-amd64.iso ubuntu-18.04-desktop-amd64.iso
$ zsync http://releases.ubuntu.com/18.04/ubuntu-18.04-desktop-amd64.iso.zsync
```

Потом запишите новый образ на диск или USB.

2 Загрузите носитель live

Установите свой диск или USB-носитель в соответствующий разъем и загрузите свой ПК. Официальный образ заработает с безопасной загрузкой (имейте в виду, что наш диск этого не сделает), но поиск магической клавиши вызова меню загрузки или настроек UEFI может вовлечь пробы и ошибки. Обычные опции — F2, F10 и Del. Пользователям Mac надо удерживать C. Выберите оптический диск или USB, затем пункт Try Ubuntu without installing [Попробовать Ubuntu без установки].

3 Тестирование и установка

Проверьте как можно больше своего оборудования в среде live (дополнительные дисплеи, всякое беспроводное оборудование и принтеры). Кроме того, удостоверьтесь, что вам всё понравилось — иначе нет смысла рисковать своей системой. Если всё хорошо, нажмите на значок установки на рабочем столе — инсталлятор сделает установку вместе с (или поверх) Windows или иных систем Linux. Если у вас более экзотическая структура, выберите опцию Something else [Другое].

Гноме для вас

Unity ушел, Gnome пришел. Несмотря на его, по мнению некоторых, полосатую историю, Gnome — мощный и полнофункциональный рабочий стол.

Возможно, вы слышали о Gnome всякие ужасы: что эта гипер-современная среда рабочего стола более подходит для телефонов, чем для ПК, что она заставляет учить трудные для понимания сочетания клавиш, что Gnome нельзя настроить, что всё написано на JavaScript... список можно продолжать. И хотя ничто из этого не является в полной мере правдой, перспектива изучать целую новую среду рабочего стола может, как минимум, пугать.

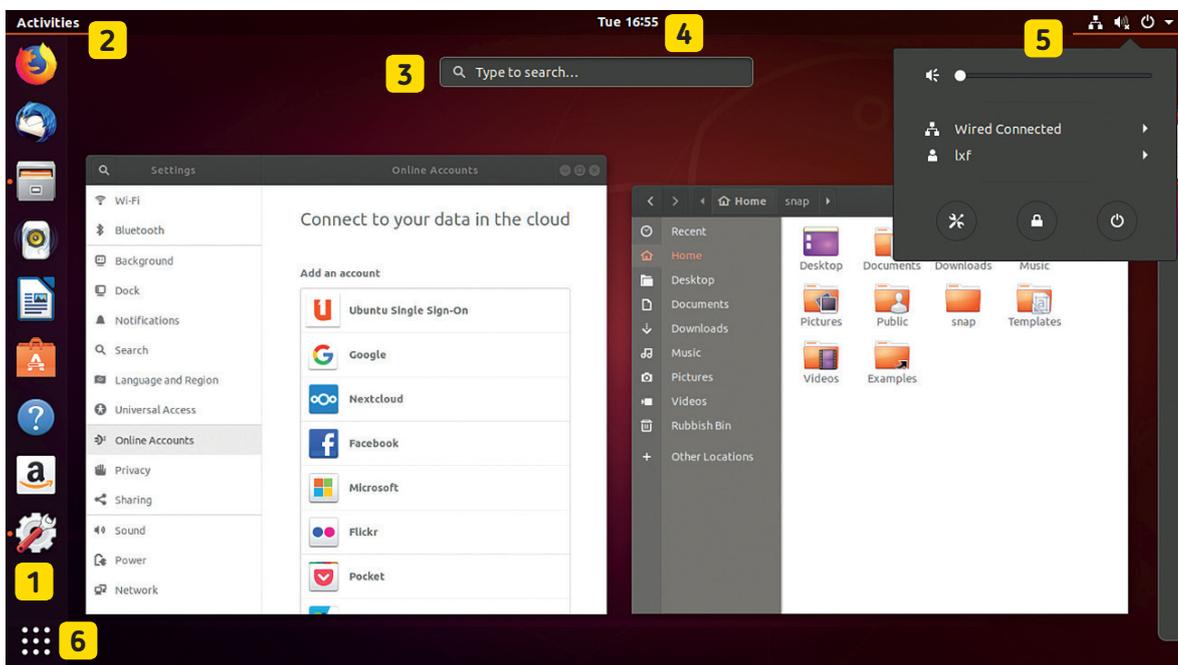
Но не волнуйтесь — Canonical немало потрудились, делая Gnome удобным для тех, кто привык работать в Unity. Есть привычная цветовая схема, индикаторы приложений, шрифт подписей и украшения окон и, конечно, характерная панель запуска слева. При желании даже можно разместить значки на рабочем столе.

Притом в Canonical очень постарались не делать свои настройки инвазивными для основы рабочего стола: всё это достигается

через стандартные расширения и темы Gnome, а не скачивание затанных библиотек. Это обеспечивает совместимость с экосистемой Gnome и позволяет обойтись без сторонних расширений, но и означает, что отсутствуют такие вещи, как общие меню и более продвинутые функции Unity HUD.

Частота релизов Gnome отлично сочетается с полугодовыми релизами Ubuntu, так что 18.04 включает (по большей части) самую свежую версию, Gnome 3.28, вышедшую в середине марта. Обо многих ее превосходных новых функциях вы можете прочитать в нашем разделе Hotpicks на стр. 100. Files, менеджер файлов по умолчанию в Ubuntu, основывается на версии из 3.26, но у него теперь стильный новый внешний вид (люди, только посмотрите на эту боковую панель!), привычный пользователям Unity. Не менее знакомой будет ссылка на Amazon на боковой панели. Однако мы советуем заменить ее на нечто более полезное — достойным и разумным выбором будет ярлык Терминала.

Знакомимся с рабочим столом Gnome...



1 Dock
Отсюда можно запускать и видеть работающие приложения (отмеченные точкой).

2 Activities
Если нажать здесь (или на клавишу Super), запустится вид Activities (на рисунке). Очень удобно, что он показывает предпросмотр работающих приложений.

3 Окно поиска
Начните вводить что-то в виде Activities (щелкать по окну не обязательно), чтобы осуществить поиск по Apps [Приложениям], Docs [Документам] и т. д.

4 Календарь и Область уведомлений
Нажмите здесь для управления музыкой, для просмотра запланированного (если вы подключены к такому сервису) и т. п.

5 Меню Состояния
Здесь находятся настройки сети (включая VPN), громкость и индикаторы приложений. Кнопка внизу слева запускает крайне важное приложение Settings.

6 Меню приложений
Нажмите здесь, чтобы открыть/закрыть часто используемые приложения. Вы можете увидеть все установленные приложения в селекторе внизу этого меню.



Мы рассказали об основах рабочего стола Gnome в нашей аннотации внизу стр. 30, а на стр. 32 вы найдете бесценный кладезь информации. Если вы решительно настроены вернуть Unity, установите его командой `sudo apt install unity`, но мы бы очень не советовали этого делать. В этом релизе были исправлены некоторые ошибки, однако в дальнейшем о нем вряд ли станут много заботиться. Странности будут случаться с нарастающей частотой, а приложения начнут сбоить; лучше всё-таки смотреть в будущее.

Если вы когда-нибудь использовали *Emacs* или пытались выйти из *Vim*, то оценили важность горячих клавиш. Вы можете преспокойно заниматься своими делами в Gnome и не знать ни одной из них, но всё же знание некоторых облегчит вам жизнь:

- » **Super (клавиша Windows)** Выводит окно activities
- » **Super-Право/Лево** Размещает текущее приложение слева или справа (чтобы оно полностью заняло половину экрана)
- » **Super-Вверх/Вниз** Разворачивает/Восстанавливает текущее приложение
- » **Super-PgUp/PgDown** Переключает виртуальные рабочие столы
- » **Super-Shift-PgUp/Dn** Перемещает текущее приложение на следующий/предыдущий рабочий стол

Это далеко не всё; больше вы сможете узнать на <https://help.gnome.org/users/gnome-help/stable/shell-keyboard-shortcuts.html.en>.

Раньше Gnome критиковали за отсутствие возможности настроек, и традиционно этому противопоставляется мнение, что пользователям незачем настраивать и без того хорошо разработанное. Доля истины тут есть, и имеет смысл привыкнуть к Gnome, прежде чем бросаться «исправлять» его расширениями.

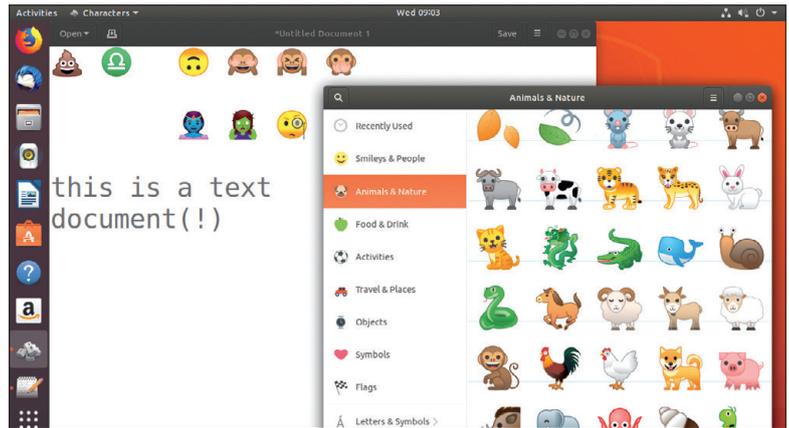
В Gnome и правда «не хватает» некоторых вещей (включая, но не ограничиваясь: минимизацию кнопок, системный лоток и строку заголовка в приложениях Gnome). Кое-что из этого можно исправить, установив инструмент *Gnome Tweaks* или расширения (см. врезку справа). *Tweaks* находится в приложении Software (там он называется *GNOME Tweaks*), и как только вы его установите, вам позволит обращаться ко множеству настроек, которые без этого были доступны только через клавиши `gsettings`.

Хорошие новости для тех, кто молод или просто любит украшать монохроматические буквы, цифры и знаки препинания цветными/выразительными/не относящимися к делу символами или их сочетанием. Ubuntu поставляется со шрифтом Noto Color Emoji, тем самым, что используется в последних версиях Android, и довольно недавними версиями Fontconfig и Cairo для их рендеринга. Их можно скопировать и вставить из приложения Characters, и они будут работать во всех приложениях Gnome (даже в Терминале!), но к сожалению, не в *LibreOffice*, хотя в нем есть поддержка некоторых черно-белых смайликов такими сочетаниями клавиш, как `⌘ y` и `⌘ g`.

Онлайн-учетки

Одна из множества впечатляющих хитростей Gnome — интеграция онлайн-учетки. Возможно, вы не хотите просматривать свои файлы на диске Google с помощью *Files*, или просматривать свои фотографии с Facebook и Flickr с помощью *Photos*, или вообще интегрироваться с любым проприетарным сервисом. Но если всё же хотите, то это можно сделать. Есть также интеграция с сервисами с открытым кодом, например, с Nextcloud.

Функция Night Light уже некоторое время присутствует в Gnome, но вы, возможно, ее не замечали. Быстрая утомляемость глаз сейчас является большой проблемой: мы слишком много времени проводим, глядя на дисплей, будь то монитор, экран телефона или очки VR. Слишком часто это происходит в ночные часы, когда голубой свет может причинить вред —



Gnome Text Editor, бывший *Gedit*, позволяет вставлять смайлики. Первый, который вставили мы, свидетельствует о зрелости нашего общества.

как физический, так и психологический. Конечно, оптимальным решением будет регулярно делать перерывы и рано ложиться спать [Ред.: — Да, Джонни!], но если так не удастся, то возможность изменить температуру цвета дисплея в нужное время может снизить напряжение на глаза. Откройте панель Settings [Настройки] и перейдите в Devices > Screen Display > Night Light [Устройства > Экран > Ночная подсветка] для настройки времени и температуры.

» УСТАНОВКА РАСШИРЕНИЙ

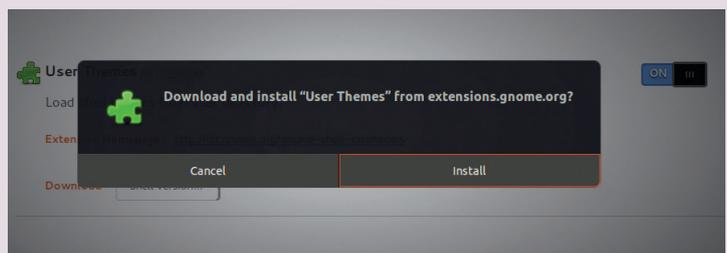
Имеется также большой выбор расширений Gnome, обеспечивающих новые функции. Нужно только сначала преодолеть две небольших сложности. Расширения Gnome устанавливаются через сайт <https://extensions.gnome.org>. Прежде чем вы сможете им воспользоваться (как говорится на сайте), вам понадобятся две вещи: «встроенное приложение для обмена сообщениями» и расширение браузера.

Первое находится в пакете *chrome-gnome-shell*, доступном в приложении *software*, а второе можно найти по ссылке [Click here to install browser extension](#) [Нажмите здесь, чтобы установить расширение браузера]. Теперь весь мир расширений перед вами. Одно из самых популярных, как вы заметите, это User Themes, оно предоставляет темы с таких сайтов, как <https://gnome-look.org>, извлекая их в директорию `~/.themes`. Сделав это, вы сможете выбирать их из *Tweaks*.

Во вкладке Extensions [Расширения] в *Tweaks* вы увидите, что два уже имеются: Ubuntu appindicators и Ubuntu dock. Их нельзя отключить, поскольку они являются ключевой частью рабочего стола Ubuntu Gnome. Ubuntu dock на самом деле ответвление другого популярного расширения, Dash to Dock, которое вас, возможно, заинтересует. Набравшись храбрости, вы также, возможно, захотите посмотреть на базовую сессию Gnome. Просто выполните

```
$ sudo apt install gnome-session
```

Вы можете выбрать сессию, когда войдете. Смотри, мам, никакого dock!



Нажмите на ползунковый переключатель и на Install [Установить], и расширение обретет жизнь.

Версии и ветки

Не расстраивайтесь, если Gnome вам не подходит. Есть и другие разновидности рабочих столов, и другие дистрибутивы на базе Ubuntu.

Сначала вышел Ubuntu (2004 г.), а потом, через год — Kubuntu, поскольку фанаты KDE хотели иметь систему Ubuntu с любимым рабочим столом. Затем появились другие разновидности: у некоторых их них в этом цикле нет релиза, а некоторых больше нет (например, Ubuntu Gnome). Но вот вам обзор наших фаворитов.

Ubuntu Budgie

Budgie — многообещающий рабочий стол дистрибутива Solus OS и последнее прибавление в семействе Ubuntu. Budgie стремится стать «понятным и мощным рабочим столом без ненужных излишеств». Одна из функций, которыми он более всего гордится — боковая панель Raven, позволяющая настраивать элементы,

не выискивая их в апплетах настроек. В этом релизе она по умолчанию скрыта, но к ней есть доступ через значок вверху справа. По умолчанию на панели Raven размещаются календарь, кнопки управления звуком и область уведомлений, но можно добавить и другие апплеты.

Ряд апплетов строки заголовка активирован по умолчанию, в том числе заметки на рабочем столе, управление ночным светом и *Caffeine*, что может остановить нежелательное опустошение экрана. Budgie включает прекрасный терминал *Tilix*, который преднастроен и предлагает выпадающий ярлык (F12). Здесь есть не менее прекрасный клиент электронной почты *Geary*. Любителям виджетов на рабочем столе Budgie позволит настроить эти виджеты прямо из окна приветствия.

В настоящий момент Budgie основан на Gnome, и в стандартном релизе Ubuntu включает множество тех же самых приложений, а также новые функции Gnome 3.28. Однако всё меняется, и Budgie 11 перейдет на Qt, чего мы ждем с большим нетерпением.

Ubuntu Mate

Mate начинался как ответвление Gnome 2 и был настоящим убежищем для тех, кто тосковал по временам каскадных меню, верхних и нижних панелей и явному отсутствию ярких эффектов рабочего стола. Однако внешний вид Mate по умолчанию изменился, что повергло некоторых пользователей в шок.

Именуемый Familiar, новый внешний вид избавился от былых меню Applications, Places и System. Но не бойтесь, старый вид (под названием Traditional) легко восстановить. Доступны также и другие виды рабочих столов, в том числе Netbook, Mutiny (ха-ха, дань Unity) и Cupertino (с доком Plank и имитирующий взаимодействие в стиле macOS).

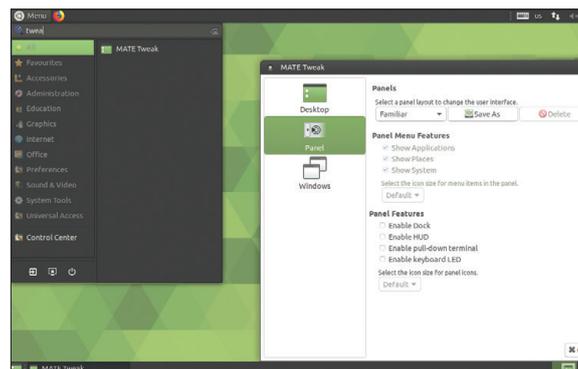
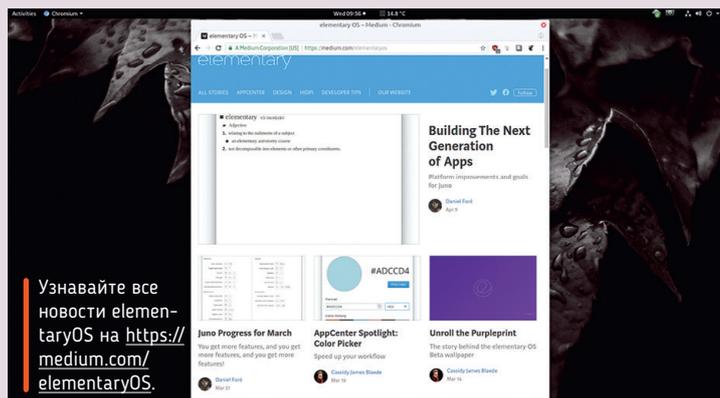
Mate отнюдь не против современных новшеств, и одно из самых интересных изменений с последнего Ubuntu — новое меню Brisk. Оно ставит Mate на равных практически со всеми остальными средами рабочего стола, поскольку дает возможность мгновенного поиска. Указанное меню можно вызвать также нажатием на клавишу Super. Как и Kubuntu, Ubuntu Mate включает опциональное Global Menu, облегчая жизнь тем, кто привык к этой функции в Unity. Дистрибутив заходит еще дальше и предлагает

» ШИРОКАЯ ЭКОСИСТЕМА UBUNTU

Помимо официальных версий, Ubuntu служит базой для ряда популярных дистрибутивов-производных, и многие из них скоро выпустят обновления. Самый популярный — Linux Mint, и мы с нетерпением ждем Mint 19, надеясь на его выход в июне. Нас впечатлил Mint 18 (см. **LXF214**), где мы увидели, как Mint начал отходить от приложений Gnome, запуская собственные, независимые от рабочего стола X-apps. Целью является избежание новых стилей GTK3, особенно строк заголовков, не соответствующих традициям.

Симпатичная elementaryOS повысит версию с 0.4 до 5.0, надеемся, в не слишком отдаленном будущем. Это разрушает наши представления о нумерации версий, зато отражает растущую стабильность и зрелость elementary. Как и в Mint, команда работает над следующим поколением приложений — все они написаны на Vala, и, в отличие от Mint, придерживаются современной парадигмы дизайна: панели заголовков и пр. Старый текстовый редактор, *Scratch*, вырос в редактор кода *Code*, который приводит нас в особое восхищение.

Если вы хотите протестировать самые последние технологии KDE, наилучшим способом сделать это будет, вероятно, через KDE Neon. KDE Neon — не дистрибутив в привычном смысле этого слова, поскольку включает только блистательные новые программы KDE. Это приводило к небольшой странности, если заметить винтажное ядро и репозитории из 16.04 LTS (идея заключалась в стабильности базы под всеми частями Plasma). Однако это исправят, по крайней мере, временно, при переходе на 18.04.



К счастью, в Mate предустановлен инструмент *tweak*, позволяющий выбрать из множества его раскладок.



HUD (Head Up Display), дающий возможность поиска по опциям меню при удержании клавиши Alt.

Kubuntu

KDE Plasma — потрясающий рабочий стол, который умудряется оставаться современным и при этом сохранять традиционную метафору рабочего стола. Реализация Kubuntu добавляет ряд приятных особенностей, со стильной темной темой и удобными ярлыками. Если у вас достаточно новая видеокарта, то все красивые эффекты (которых явно много) получат аппаратное ускорение и не будут тормозить вашу работу. Kubuntu использует Plasma 5.12 LTS, вышедший в феврале. Но PPA будут доступны для пользователей Kubuntu, желающих отважно установить библиотеки поновее.

Ранее KDE считался самым прожорливым по ОЗУ рабочим столом, но в наших тестах эта бесславная корона ушла к Gnome (хотя это не так уж важно сейчас, когда 16 ГБ памяти — почти стандарт). KDE отлично настраивается, предлагает большой выбор программ запуска («классические» меню, меню «с выдумкой» по умолчанию и полноэкранный запуск а-ля Gnome). При желании даже можно настроить панель меню Global (чтобы меню приложений появлялись в верхней панели, а не в окне приложения; это функция Unity).

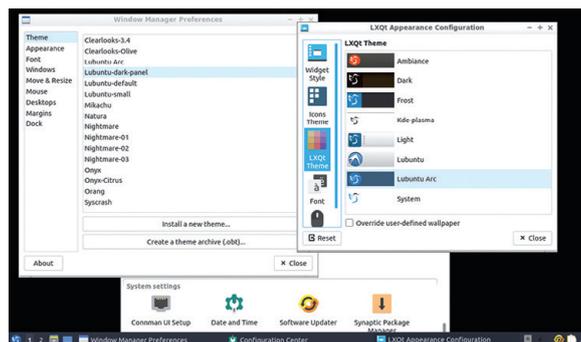
Lubuntu

Lubuntu — наименее алчный к ресурсам из всех Ubuntu. Он устанавливается даже на Pentium II с 128 МБ ОЗУ и 2 ГБ на жестком диске, но для повседневного использования рекомендуется 512 МБ ОЗУ. Основан Lubuntu на устаревшем инструментарии GTK2, и это проблема, потому что многие приложения больше не делают сборок с ним без густого латания, и эта тенденция будет продолжаться в течение пятилетнего периода поддержки.

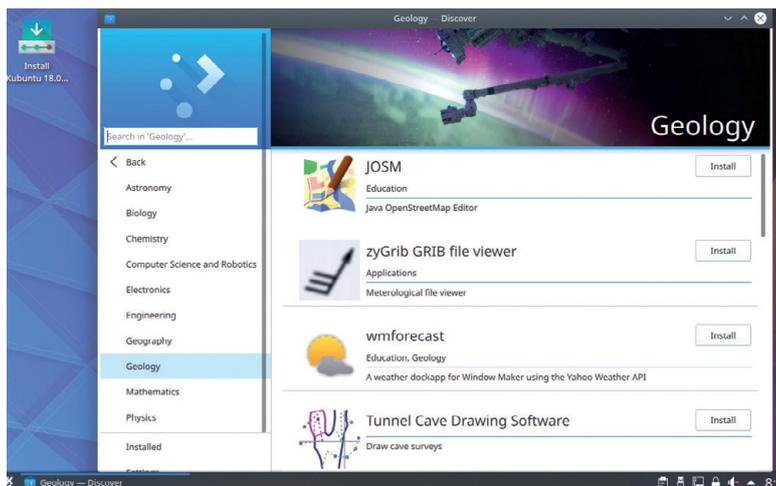
Lubuntu находится в процессе перехода на рабочий стол LXQt на Qt5, но это процесс сложный, и команда не хочет загружать нечто недоделанное (и потом пять лет это поддерживать). Однако работа ведется, и образ Lubuntu-next доступен для тех пользователей, кто жаждет взглянуть в будущее. Он будет поддерживаться девять месяцев, и за это время проект станет более пригодным для широкой аудитории.

Lubuntu Next потребует чуть больше усилий, чем его предшественник, и его требования к памяти и объему диска чуть более высокие, но должен нормально работать на компьютерах, собранных в последние десять лет. Благодаря большой работе по отделению вспомогательных библиотек KFrameworks от остального KDE, дистрибутив не перегружается специальными библиотеками KDE, как это традиционно делалось при установке программ Qt4.

LXQt 0.12 вышел в октябре 2017 г., и на нашем запыленном оборудовании мы протестировали его с отличными результатами.



Lubuntu использует рабочий стол LXDE, который работает на менеджере окон Openbox, чтобы получить рабочий стол без ненужных излишеств.



Его легковесность не означает, что он пожертвовал чем-то в плане стиля или функциональности.

Xubuntu и Ubuntu Studio

Бывший когда-то единственной (полнофункциональной) альтернативой Gnome и KDE, рабочий стол Xfce, использовавшийся в Xubuntu, к сожалению, затмили такие рабочие столы, как Mate и LXQt. У проекта не было новых релизов с февраля

KDE Plasma имеет собственное программное приложение, Discover. Раздел Geology рулит — сами посмотрите!

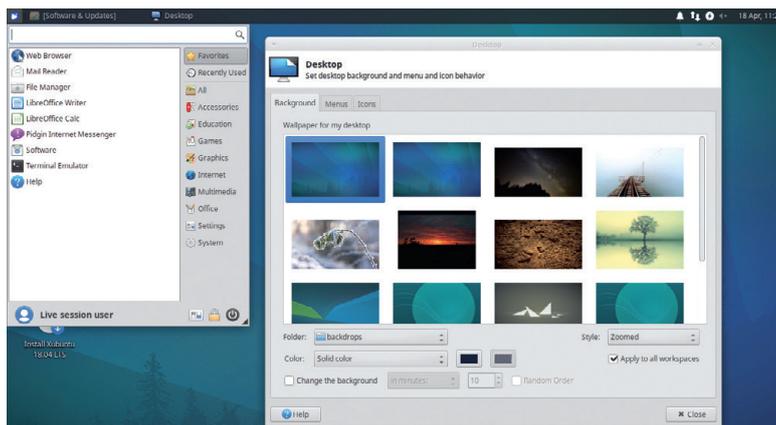
KDE ВОЗВРАЩАЕТСЯ В СВЯЩЕННЫЕ КНИГИ

«Ранее KDE считался самым прожорливым по ОЗУ, но в наших тестах эта бесславная корона ушла к Gnome.»

2015 г. Но он далеко не в застое (посмотрите на активность на <https://git.xfce.org>), медленный переход на GTK3 и долгожданный релиз 4.14 еще в процессе, и немалое количество компонентов больше не зависят от GTK2.

Xfce также является рабочим столом Ubuntu Studio, разновидности для тех, кто занимается мультимедиа и творчеством. Этот релиз — не LTS, но он поддерживается девять месяцев, и ведутся работы по оживлению сообщества Studio. Следующий релиз, 18.10, надеется всё немного изменить. Тем не менее, этот релиз демонстрирует отличные инструменты: JACK, Blender, Darktable, Ardour и многие другие. Это единственный дистрибутив, где вы найдете все эти программы предустановленными.

Трудно написать подобный материал и не упомянуть предлагаемых вариантов фона рабочего стола — вот вам роскошная подборка Xubuntu.



Ubuntu следующего поколения

В этом релизе множество интересных новых технологий (помимо Gnome) — осматривайтесь и выясняйте, что есть что...

Ядро 4.14 тоже является релизом LTS, поэтому команда ядра будет поддерживать его как минимум пять лет. Планировалось использовать его для Ubuntu, но поскольку 4.15 вышел в январе, и работа с постоянно развивающимися заплатками KPTI для Spectre/Meltdown в некотором смысле проще, чем с обратно портированными решениями, было решено использовать 4.15 для Ubuntu.

Ядро 4.15 было одним из самых трудоемких релизов за довольно длительное время. Помимо большого количества заплат против Spectre и Meltdown, оно также включает массу потрясающих новых функций. После серьезной переработки появился уровень AMD DC (Display Code), и это означает, что владельцы карт

его при желании можно употреблять из удобного приложения Software and Updates. Требуется учетная запись Ubuntu Single Sign On (некогда используемая для ныне покойного сервиса хранения Ubuntu One), которую можно зарегистрировать, следуя инструкциям. Эта учетная запись также позволит вам заходить на Ubuntu Forums (<https://ubuntuforums.org>), чтобы разместить там любой возникший у вас вопрос.

Live kernel patching [латание ядра на лету] существует уже не первый год (впервые мы рассказали о нем в **LXF200**), и оно очень удобно на случай, когда вы не можете загрузиться, но хотите применить какие-нибудь обновления ядра, особенно если они касаются безопасности. Однако важно отметить, что в общем livepatching не отменяет перезагрузки. Например, обновления Systemd часто требуют перезагрузки, чтобы вступить в силу, и обновления рабочего стола в основном не начнут работать, пока вы, как минимум, не выйдете и снова не войдете. Главные обновления ядра (например, переход на уровень выше) по-прежнему требуют перезагрузки, и по нашему опыту, снап темы сообщества (см. ниже) тоже не покажет без нее новую сессию. Считайте, что вас предупредили.

Одно немного спорное новшество в этом релизе — сбор статистики пользователя. Данные по установленным пакетам собираются через пакет popularity-contest, который настраивает процедуру cron для ежедневного обновления пакетов. Этот пакет долгое время был опциональным в Debian, и всё анонимно и отправляется через HTTPS. Как и в других операционных системах, если что-то пошло не так, вы можете отправить в Canonical сообщение про собой. В некоторых случаях это переведет вас прямо в соответствующее сообщение об ошибке, где вы, при известном везении, сможете найти временный обходной путь.

Обновление тем

Ambiance, долго бывшая темой Ubuntu по умолчанию, для 18.04 была поднастроена, придав Gnome еще большее сходство с Unity по виду и ощущениям. Немного встряхнули и более светлую тему. Проект Ubuntu Communitheme (см. <https://didrocks.fr/2017/11/09/welcome-to-theubuntu-bionic-age-a-new-ubuntu-default-theme-callfor-participation>) находится в разработке, но пока не готов выйти на суд широкой публики. Те, кто не боится неустраивающих краев, могут попробовать его благодаря тому, что он распространяется в виде снап-пакета. Его можно установить с помощью `sudo snap install communitheme` и испытать, перезапустив систему и выбрав сессию communitheme в окне приглашения. Тема сообщества началась с Adwiate, темы Gnome по умолчанию, и предназначена она стать полной заменой Ambiance и данью покинутому Unity 8. Читайте больше в блоге разработчика Canonical Дидье Роша [Didier Roche] (<http://bit.ly/ubuntu-bionic>).

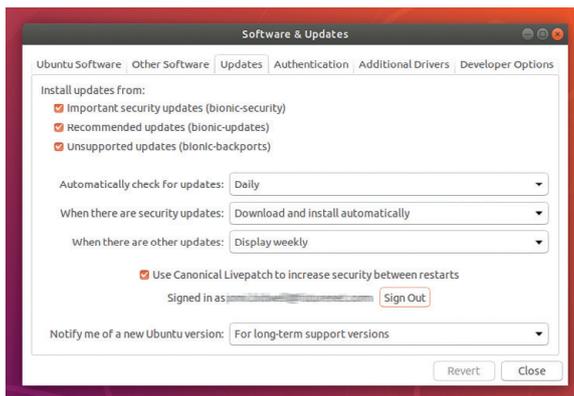
И кстати о снап-пакетах... Снап-пакет — это излюбленный подход Canonical к составлению пакетов независимых от дистрибутива приложений типа Flatpak и AppImages. Помимо того, что он облегчает жизнь разработчикам (объединяя с приложением

СОМНЕНИЯ? ПЕРЕЗАГРУЖАЙТЕСЬ!

«В общем livepatching не отменяет перезагрузки. Например, обновления systemd часто требуют перезагрузки, чтобы вступить в силу.»

серий Vega и Raven Ridge (включая 2200G, о которой мы рассказали на стр. 17) могут использовать драйвер с открытым кодом AMDGPU, не нуждаясь в индивидуально настроенном ядре. Что также позволяет этому драйверу работать с HDMI-аудио на других графических процессорах от AMD. Владельцы ноутбуков могут заметить увеличение срока жизни батареи благодаря улучшению управления питанием SATA. Canonical вложил немалый труд в поддержку ядра (в конце концов, это сердцевина его облака, сервера и предложений IoT), и мы можем рассчитывать на то, что функции из будущего ядра обратно портируют туда, даже после того, как сама команда ядра прекратит работу над серией 4.15.

Сервис Ubuntu Livepatching интересен скорее администраторам сервера, чем пользователям настольных ПК, но теперь



Настраивайте обновления отсюда. Если вам нужен электронный адрес Джонни, то он указан в конце журнала.



все требуемые библиотеки) и составителям пакетов (полностью устраняя их из уравнения), эти форматы могут предложить некоторые преимущества в области безопасности благодаря использованию технологии контейнеров для увеличения степени изолированности.

Первым делом в приложении *Software* показаны приложения Snap, но, немного поискав, вы всё равно найдете программы, распространяемые в традиционном формате Deb. При установке программ из «магазина» *Software* процесс одинаков для всех и не зависит от формата пакетов. Традиционный метод `apt install` по-прежнему в порядке вещей для Deb, но для установки Snaps вам придется выучить несколько новых команд командной строки. Выше мы видели, как использовать Snap темы, но у них есть и другие хитрости. Например, чтобы переключиться на ветвь `dev` в `communitHEME`, используйте

```
$ sudo snap refresh communitHEME --edge
```

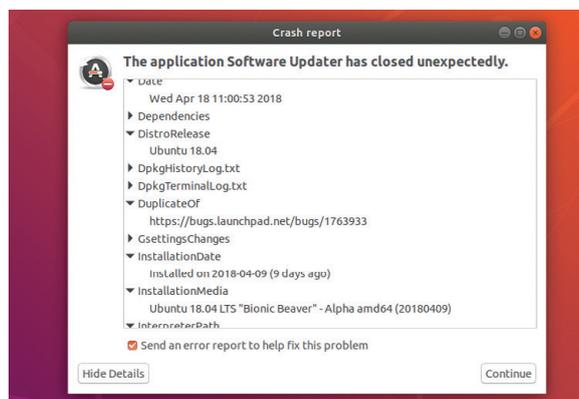
Замените `--edge` на `--stable`, чтобы переключиться на стабильную ветвь. Snap будут автоматически обновляться *Software Updater*, но не *Apt*. Чтобы обновить их из командной строки, скачайте `snap refresh`.

Ubuntu Mate 17.10 сделал прорыв, предоставив `pulseMixer` в виде Snap, поскольку он не включался в пакеты для Ubuntu или Debian. Эта тенденция теперь уже проложила путь в основной дистрибутив, который включает ряд предустановленных Gnome Snaps: *System Monitor*, *Logs*, *Calculator* и *Characters*. Вы можете увидеть их смонтированными в директории `/var/lib/snapd` через команду `mount`.

Wayland был сервером отображения по умолчанию в Fedora с версии 25. Поговаривали, что он станет им же в Ubuntu, но после экспериментов с ним в 17.10 по умолчанию вернулся старый добрый X.org. Тем не менее сессия Wayland доступна (просто нажмите на значок с шестерней), и наше краткое тестирование показало, что она работает хорошо. Даже если вы используете проприетарный драйвер Nvidia, который реализует Wayland через EGLStreams, Gnome должен работать хорошо. Возможно, вы слышали истории о том, что скриншоты, запись с экрана и удаленные сессии рабочего стола с Wayland не работают, но это только полуправда. Всё это работает, если их правильно поддерживает композитор и приложения записи правильно разговаривают с композитором. К счастью, композитор Gnome (*Mutter*, он же и менеджер окон) обеспечивает и то, и другое, и даже имеет встроенную программу записи с экрана, которую можно включить с помощью `Ctrl + Alt + Shift + R`.

Конец 32-битным?..

Как и предсказывалось, это первый релиз Ubuntu LTS, не предлагающий 32-битного установщика. Это не означает, что ваше



Ubuntu позволяет отправлять отчеты о сбоях, помогая команде понять, что не работает. Опционально они могут также собирать анонимные данные об использовании.



допотопное оборудование не может заполнить Ubuntu 18.04, или даже что Ubuntu не будет хорошо на нем работать (если только у вас хватает памяти); это просто означает, что у вас не будет чудесного готового к работе установочного диска. Если у вас уже работает 16.04 на 32-битном оборудовании, то вы должны обновиться до новой версии без всяких проблем.

Если вы хотите сделать чистую установку, есть парочка опций. Вы можете взять одну из разновидностей: все они предлагают 32-битные средства, но лучшими кандидатами будут Lubuntu, Ubuntu Mate или Xubuntu. Если вы правда хотите запустить Gnome на 32-битном оборудовании, можете установить его поверх любого из них через пакет `ubuntu-gnome-desktop`. Другим подходом будет начать с ISO сервера i386 и затем скомпилировать оттуда свой рабочий стол. Это будет хорошая опция для создания легковесного рабочего стола с помощью Openbox, i3 или чего-то подобного. Забавно, что в следующем номере у нас есть статья о том, как это сделать. Надеемся увидимся с вами там! **LXF**

SDLpop, версия *Prince of Persia* с открытым кодом, доступна в виде Snap.

» ВНЕ НАСТОЛЬНЫХ ПК

Мы сосредоточились на настольных ПК, однако Canonical зарабатывает деньги не на них, и это далеко не вся история Ubuntu.

Ubuntu Core — это крохотный дистрибутив Canonical, разработанный для сетевых устройств, в частности, для объектов IoT и контейнеров. Core использует то же ядро, что и настольный и серверный Ubuntu, и задействует Snap-пакеты для обеспечения гладкой и безопасной установки и обновления. Canonical надеется, что Ubuntu Core станет популярным для компьютеров Edge (или Fog). Это граница между облачными и реальными устройствами, где устройства находятся на грани маломощных (например, сенсорных) сетей и направляют их данные в облако.

Иногда «меньше» значит «больше», и это высказывание никогда не было более верным, чем применительно к контейнерам, встроенным системам и IoT. В этой связи Canonical сейчас предлагает минимальный образ — с учетом того, что он занимает менее 28 МБ, это и правда минимум. Его не надо путать с минимальной настольной установкой, о которой мы говорили ранее: она всё же представляет такую роскошь, как рабочий стол, и такое излишество, как программа установки. О нет, это всего лишь `tag`-архив файловой системы, которая мало что дает помимо рабочей оболочки.

Это идеальная стартовая точка для ситуаций, когда есть ограничения и нужно только самое необходимое и ни байта больше. Однако это нельзя устанавливать на голое оборудование, потому что оно не может загрузиться. Многие официальные образы Docker (включая, например, *Apache* и *MariaDB*) в настоящее время используют в качестве базы крошечный дистрибутив Alpine Linux. При 8 МБ он намного меньше предлагаемого Ubuntu, но последний может быть удобнее для сторонников контейнеров, готовых принять лишние мегабайты, чтобы использовать более знакомые *Apt* и *glibc* (в противоположность *apk* и *musl*).

ПРОГРЕССИВНЫЕ СЕССИИ

Джонни Бидвелл
беседует о прогрессивных
web-приложениях
с соучредителем
компании Eponic
Томасом Сигдестадом.

Томас Сигдестад [Thomas Sigdestad] — технический директор и соучредитель компании Eponic, трудное детство которой началось в гараже в центре Осло. У нее была смелая идея дать людям нечто полезное во времена бумага интернет-компаний, и сегодня это всё еще очень большая часть стратегии. В Лондоне мы пересеклись с Томасом на конференции O'Reilly Software Architecture 2017, чтобы обсудить будущее кросс-платформенных приложений: прогрессивные web-приложения. Возможно, вы не слышали об этом, но, скорее всего, если вы недавно перешли куда-то недалеко с основного сайта, вы уже это использовали.

Новые технологии позволяют web-браузерам делать всевозможные вещи — такие, которые до сих пор строго охранялись нативными приложениями. Это может означать много интересных разработок: появление действительно кросс-платформенных приложений, конец бедствий магазинов приложений; возможно даже, Linux станет практичным решением для людей, связанных запросами приложений для проприетарных ОС. Прекрасный новый мир ждет, так что читайте дальше...

LXF: Привет...

Томас Сигдестад: Итак, вы те самые люди, которые делали *Amiga Format*? Я всё еще помню, как с нетерпением ждал дисков и первый обзор последней игры. Это было довольно дорого для Норвегии, помнится, £7 или £8.

LXF: Это мы... ну, не мы лично, а наша компания. Признаться, я любил этот журнал. Моя мама обычно злилась, что я трачу £3 на журнал. Она не понимала удивительной ценности, которая в нем была. Дело было не только в качественной журналистике — диск обложки был грандиозным событием: 880K PD-игр и демо-версий, которыми я был занят, ну, недели четыре. Ах, ностальгия, лучше не заводи меня. Как вы попали в Linux?

ТС: Когда я учился, купил набор для установки Red Hat в университетском книжном магазине. Кучу времени потратил на его изучение. Помню, думал — это реально круто; а что, по-моему, и до сих пор круто — это сетевая загрузка.

Ты просто запускаешь машину, а она отключается и извлекает образ. В 1997 г. это было потрясающе.

LXF: Я пользовался Red Hat в начале 2000-х. Как только у меня появился компьютер, способный загрузить Windows XP, я понял, что мне это не нравится. Нисколько. Сейчас я не такой воинственный, но с тех пор Linux стал моей основной ОС.

ТС: Забавно, что наш продукт называется Eponic XP. XP — сокращение от Xperience Platform, но это не связано с Windows XP. Когда мы выпустили его



Томас стал соучредителем компании Eponic в 2000 г.

два года назад, то представили его как «операционная web-система». Создавая приложение для Android, вы знаете, как записать на диск и воспроизводить графику и тому подобное. Других зависимостей нет, просто версия для Android. У нас была такая же идея для серверов: когда вы создаете приложение Eponic, вам надо лишь уточнить, какую версию XP вы использовали. Это концепция ОС на гораздо более высоком уровне — подход

идея — типа в духе Билла Гейтса — что успехом Microsoft обязан копированию IBM. Мы сделали примерно то же с нашим сервером приложений и базой данных. Изначально у нас были клиенты, инвестирующие в такие архитектуры, им были интересны наши решения. И, камень за камнем, за пять лет мы построили наш бизнес. Мы сделали четкую CMS и поняли, что вещи во многом изменились. У нас была команда, работающая

ЕСЛИ ЭТО СРАБОТАЛО ДЛЯ MICROSOFT...

«У нас была идея — в духе Билла Гейтса — что успехом Microsoft обязан копированию IBM. Мы сделали примерно то же с нашим сервером приложений и базой данных.»

с распределенным сервером. Мы зашли на рынок с именем XP по маркетинговым соображениям, но, возможно, через несколько лет термин «web-операционки» приживется.

LXF: Расскажите о своей компании Eponic.

ТС: Мы начали работать в 2000 г. Хотели стать компанией-разработчиком ПО, поэтому около года работали, создавая наш web-портал, а затем запустились и на удивление быстро получили интересных клиентов. Linux мы использовали для различных целей и в различных вариантах с первого дня. Для нашего хостинга мы всегда использовали Linux.

Всё то, что сделало нас уникальными, мы разработали при посредстве Java. У нас была

над стеком приложений, команда, работающая со стеком баз данных, и еще одна команда, работающая над стеком сайтов. Но в конечном итоге, все они работали над созданием опыта взаимодействия и должны были как-то связывать эти стеки между собой. Затем мы поняли, что это можно сделать иначе.

Инвестируя в CMS, приходится иметь полный стек, включая базу данных и всё прочее. Поэтому мы построили всё с нуля, в цельном комплекте ПО. И CMS — верхний уровень этого стека, так что у нас есть платформа приложений и CMS как дополнительная функция стека. Но обычно именно это нужно нашим клиентам. Мы могли бы сделать много интересных решений, но решили сосредоточиться на трех ключевых областях:

цифровом опыте, прогрессивных web-приложений и услугах движка.

LXF: Давайте поговорим о прогрессивных web-приложениях. Простите мое невежество, но что это вообще?

ТС: Краткое пояснение — это web-приложения, в которых вы работаете как в нативном приложении, но построены они на web-технологиях. Вы можете видеть пример на <http://officeleague.rocks>. Это приложение foosball, которое позволяет любому создать свою лигу. Оно будет работать в любом современном браузере, на настольном ПК или мобильном устройстве. Всё управляется JavaScript, поэтому не перезагружается условная страница, которая каждый раз загружает код приложения. Вы получаете «рейтинг Эло», как в шахматах, и можете смотреть игры вживую, что очень круто.

Круто — если, например, я играю на своем телефоне и включаю режим самолета, тогда web-страница обнаруживает, что я в автономном режиме; но я всё равно могу продолжать игру. Приложение всё записывает, и когда я возвращаюсь в онлайн, приложение отправляет данные обратно на сервер. Помимо этого, при первом запуске приложения вас спросят, хотите ли вы добавить его на свое устройство. Если вы соглашаетесь, то получаете иконку, как для обычного приложения. И как в обычном приложении, вы можете посмотреть информацию о нем

в настройках приложения Android или удалить его. У вас есть доступ ко множеству всяких аппаратных функций, чего вы обычно не ожидаете от web-приложения.

Таким образом, прогрессивное web-приложение не является определенной частью технологии, это скорее схема того, что могут делать эти приложения и как они работают. Подводя итог: они ощущаются как нативные приложения, и они кросс-платформенные, так что их надо лишь один раз закодировать вместо того, чтобы создавать для Android, iOS, web и Windows отдельно. Также они отзывчивы, поэтому могут иметь экраны разных размеров, могут работать в автономном режиме, т.е. это устойчивость независимо от состояния сети. Причина, по которой они называются прогрессивными, заключается в том, что основные подсистемы этого приложения, например, foosball, работают на iPhone, даже если не поддерживают автономный режим или ряд других функций. Т.е. вы можете получать прогрессивные улучшения вашего приложения в зависимости от возможностей клиента.

LXF: Как насчет ускорения видео? Используют ли они WebGL или другие трюки, чтобы мы получили приятный процесс?

ТС: Да, в принципе они могут использовать любую технологию, которую ваш браузер способен употребить на вашем устройстве и вашей платформе. Используется это или нет — зависит

от разработчика. Если приложение создавалось ради бешеной производительности, то можно применять WebGL или даже WebAssembly. Это только один случай применения, хотя это не всех волнует. Большинству бизнес-моделей для создания приложений нужны только стандартные web-технологии.

LXF: Легко ли писать такие штуки?

ТС: Разработчикам довольно просто это использовать, и мы полагаем, что года через два разработка нативных приложений станет нишевой и будет применяться только по особому запросу. Например, Google только что заявил об API WebUSB и Payment Request для Chrome (см. https://developer.chrome.com/apps/app_usb и <https://developers.google.com/web/fundamentals/payments/>). Это поразительно. Например, если вы PayPal, то можете создать web-приложение или нативное приложение и зарегистрироваться как поставщик платежей на данном устройстве. Затем сайт может просто вызвать API запроса платежа, и пользователю выдадут список поставщиков платежей — PayPal, Apple Pay или что-то еще; поэтому устраняется значительная сложность при осуществлении реальной транзакции.

Аналогично, WebUSB открывает другие новые возможности: всё, от управления камерой (каждый ее элемент), микрофон, разрешения доступа, всё это встроено в Web. Сеть всегда была задействована за кулисами, но теперь она вышла на первый план, это абсолютно жизнеспособная альтернатива нативным приложениям.

LXF: Магазины приложений стали обычным стандартным способом получения приложений. Даже инструмент Gnome Software в значительной степени соответствует этой норме. Как это работает для прогрессивных web-приложений?

ТС: Ну, когда вы посещаете web-страницу, никакой магазин приложений не задействован, поэтому целый шаг «покупка/загрузка от Google Play» или любой другой системы здесь просто пропускается. Так что это хорошо и для разработчиков, и для пользователей, поскольку ускоряет работу. Вероятно, Twitter — лучший пример прогрессивного web-приложения. В нем 3% от объема нативного приложения для Android. Обновления не нужны: посещая сайт, вы всегда запускаете свежую версию.

Таким образом, вы получаете лучшее из традиционного мира web-приложений в сочетании с лучшим в мире нативных приложений. Речь идет только о том, чтобы web-браузеры поддерживали необходимые функции. Эта инициатива больше всего продвигается Google через Chrome и Chromium. Firefox тоже отлично справляется (см. <https://developer.mozilla.org/en-US/Apps/Progressive>). Microsoft тоже задействована, так что

» КАК ENONIC ЭКОНОМИТ ВАШЕ ВРЕМЯ

LXF: Итак, вы основали Enonic в период краха dot-com'ов, верно? Но у вас была хорошая идея, в отличие от многих компаний, которые просто вставляли слово «Интернет» в корявые бизнес-планы.

ТС: Да, мы начали летом 2000 г., а через месяц весь мир рухнул. Мы обратились прямо к платформе, что не так просто. Отчасти нам повезло, поскольку мы думали «Давайте сделаем всё чертовски простым» и хотели сделать CMS настолько удобной, чтобы любой мог ее использовать.

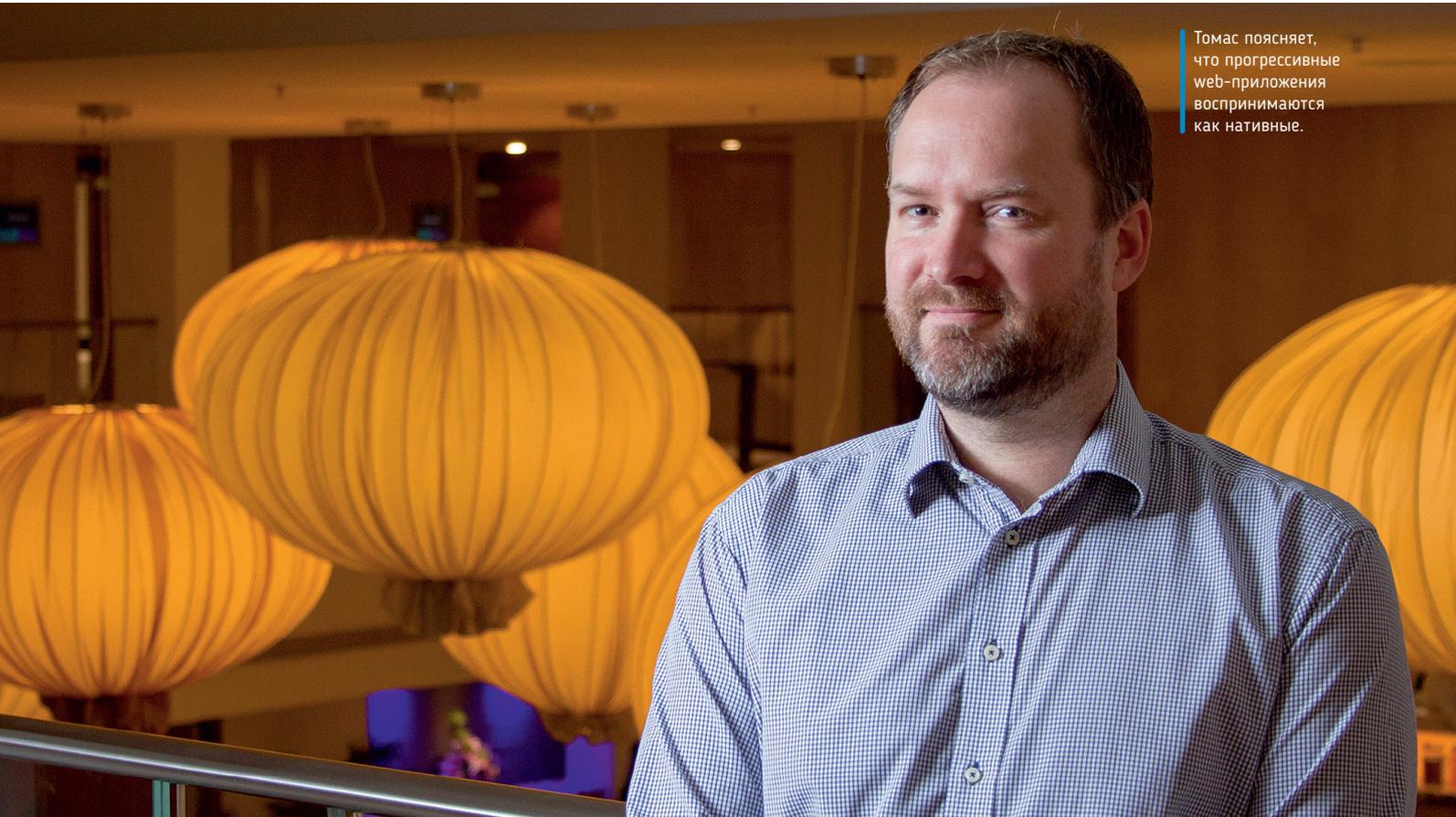
Мы должны были предоставить пользователям ввод на основе форм, а для этого требуется довольно обширное кодирование. Оказалось, что ни одна мелкая компания не горела желанием вкладываться в кодирование такого уровня, но этого хотели крупные компании; а поскольку мы уже были вовлечены в Java, они были нам рады. Мы были очень молоды, когда начинали, но создали вполне, как мы думали, годный продукт — и в то же время

толком не знали, кому наш продукт нужен. Теперь мы представляем это лучше, но всё меняется очень быстро, особенно с облаком.

Сегодня наше предложение по существу «У вас есть идея. У нас есть платформа. Сможете запустить ее в любом облаке, если построите на нашей платформе». В типовых случаях мы заменяем SQL/NoSQL, поиск, движок приложения и по крайней мере часть web-сервера. У нас есть система идентификации, вы можете подключить другие, и у нас есть наша CMS. Поэтому мы предлагаем, может быть, пять систем в одном.

Традиционно люди привыкли сначала всё это складывать, прежде чем писать хоть строку кода. И настройка этих компонентов, чтобы они ладили друг с другом, может отнять массу времени. Если вы довольны нашей настройкой, вы можете сберечь это время. И вам не надо беспокоиться о сохранении, обновлении и масштабировании всех этих компонентов.

Томас поясняет, что прогрессивные web-приложения воспринимаются как нативные.



это еще один шанс для них встать на сторону клиента. Компания собирается иметь прогрессивные web-приложения в Windows Store.

LXF: Как насчет Linux? Предположительно, прогрессивные web-приложения могут работать там, как и везде?

ТС: Так и есть, и это интересно. Linux доминирует практически во всех других областях — облачном, мобильном (если мы разрешаем Android быть Linux), но никогда не был основной настольной ОС. А с прогрессивными web-приложениями настольный Linux может стать жизнеспособной альтернативой macOS и Windows.

LXF: Это немного похоже на то, что браузер становится новой ОС?

ТС: Да, по крайней мере, если вы думаете об этом как о рабочей среде для приложений. Они стали основой для клиентских приложений. Это немного похоже на то, что сделала Java для кодирования серверных приложений в конце 1990-х.

LXF: Забавно, в те дни, когда я видел “Starting Java...” в строке состояния Netscape, меня бросало в дрожь. Я знал: скорее всего, моя машина будет стоять на коленях, пока не загрузится

какой-нибудь апплет. Но так было на стороне клиента, и, вероятно, моих 486 МБ было мало даже по тем временам. На стороне сервера Java позволяла всевозможным вещам быть больше и быстрее. Я слышал, что именно поэтому Twitter смог так масштабироваться.

ТС: Да, можно сказать, что большинство крупных сайтов социальных сетей сейчас работает на Java. И сама Java — это хорошо, поскольку помогает

поиск, например, и это действительно помогло его масштабированию. И да, Java довольно распространена — наша платформа основана на Java. Однако мы увидели новую тенденцию, связанную с прогрессивными web-приложениями и сетью в целом: JavaScript. Когда вы создаете приложения на нашей платформе, вы кодируете на JavaScript, как на стороне сервера, так и на стороне клиента.

ШАГАТЬ В НОГУ СО ВРЕМЕНЕМ ПРОСТО

«Twitter – лучший пример. В нем 3% от объема нативного приложения для Android. Обновления не нужны: вы всегда запускаете свежую версию.»

с многопоточностью и тому подобным. А сообщество поверх Java, и особенно со стороны Linux, создает всякие интересные штуки с открытым исходным кодом, которые ее еще улучшают. Когда Twitter пошел набирать силу, то сперва поимел «детские болезни», но затем начал применять некоторые из этих вещей, чтобы улучшить свой

LXF: Это интересный момент. Многие люди не знакомы с серверной стороной JavaScript, я привык к неуклюжему демпингу JS-кода в HTML-файле, например, но такие вещи, как Node.js, меня путают. Можете ли вы объяснить, как это работает?

ТС: По сути, JavaScript был создан Netscape, чтобы сделать кое-какие хитрые трюки



в браузере. Но когда вы запускаете JS в браузере, это единственный поток UI, и обычно у вас также есть DOM (модель объекта документа), которой вы управляете. Итак, обычно вы видите: есть событие, какой-то JS, он реагирует и что-то меняет в DOM.

На стороне сервера нет DOM, поэтому там есть несколько разных подходов. Существует сам язык, и вокруг **Node.js** происходит огромное движение, которое работает почти так же, как и в UI: это единственный поток, и в итоге вы запускаете кучу небольших серверов и вам надо направлять трафик в них, что и прекрасно, да к тому же без отслеживания состояния.

Здесь тот же самый принцип: что-то приходит, начинается событие, а затем вы можете делать любые асинхронные трюки между сервером и браузером. Тем не менее, мы хотели, чтобы люди могли получать всё, что вложено в Java, всю многопоточность и способность полностью задействовать аппаратное обеспечение. Кроме того, многопоточный код проще, чем асинхронный, его проще отлаживать и т. п. Таким образом, мы позволяем вам кодировать серверную часть JavaScript: вы получаете начальный запрос, вызывается часть вашего кода, а затем запускается сам JS.

Потом, есть еще возможность под названием **commonJS**, которая, вкратце, позволяет делать вставки. Стандартный JS такого не поддерживает. Эти понятия одинаковы для клиента и сервера,

но в **node.js** у вас есть только этот единственный поток. Наша платформа имеет однопоточную модель. Хотя идея та же, что и в любом языке скриптов, вы запускаете скрипт и начинаете делать вещи программно. И можете использовать возможности всех сторонних библиотек или утилит, которые помогут вам делать свое дело.

Есть и еще одна приятная штука: когда вы кодируете клиентскую часть, в **Chrome** можно пере-

LXF: А как с техническим сообществом в Норвегии?

ТС: О, оно действительно большое. Одна из крупнейших конференций разработчиков, **JavaZone** (<https://javazone.no>), проводится в Осло. По-моему, в этом году было 2500 разработчиков. Хотя одна вещь в Норвегии немного печальна, она состоит в том, что 90% происходящего [в техническом мире] является консультированием. Правительство

ПОЧЕМУ ИМЕННО НОРВЕГИЯ

«Техническое сообщество в Норвегии большое. Одна из крупнейших конференций разработчиков, **JavaZone**, проводится в Осло. В этом году было 2500 разработчиков.»

ти в Инструменты разработчика и найти консоль отладки, которая облегчает жизнь. При разработке на стороне сервера вы часто оказываетесь во власти клиента, поэтому отладка сложнее. Но на нашей платформе можно использовать ту же самую консоль отладки, чтобы следить за кодом на сервере, и это отличная возможность для разработчиков. Платформа **Javascript**, которую мы используем в **Eonic XP**, также доступна в виде автономного проекта **PurpleJS** (<http://purplejs.io>).

и крупные компании нанимают людей в качестве консультантов. Таким образом они забирают все таланты, и культура разработки продукта менее сильна. В настоящее время есть больше возможностей для людей, способных участвовать в стартапах и т. д., но всё еще слишком много внимания уделяется консультациям.

Но ряд важных компаний ушел из Норвегии: **Fast Search & Transfer (FAST)**, теперь часть **Microsoft**; **Qt** — очевидно, немаловажная для **Linux**; **Trolltech** (куплена **Nokia**). Так что, по крайней мере, в Норвегии, успешные компании склонны кому-то продаваться. И, возможно, норвежские инвесторы не совсем осознали потенциал цифрового рынка. Но очень много людей опытных — недостатка в талантах нет.

LXF: Расскажите нам о своем выступлении.

Я его пропустил, потому что должен всё время общаться с людьми умнее меня.

ТС: Ну, я окупал прогрессивные **web-приложения** в тематическую дискуссию «Звездные войны». **Apple** была темной стороной. Такие вещи хорошо делать хотя бы немного увлекательными. Посыл состоит в том, что мы пытаемся создать платформу, которая помогает людям создавать мощные **web-приложения**. Мы считаем, что прогрессивные **web-приложения** станут массовыми, но большинство людей еще не знают об этом. Это только вопрос времени, прежде чем всё начнется. И уж точно, что для нативных дистрибутивов **Linux** это будет действительно здорово.

LXF: Потрясающе, большое спасибо, что уделите мне время. Я думаю, что Джеймс, наш фотограф, теперь собирается организовать в холле снимок героя.

ТС: «Геройский» снимок?! О нет, я же не привез свой супергеройский костюм из латекса! **LXF**



Здесь мы видим Томаса без костюма супергероя. Но всё же он героически помогает людям создавать **web-приложения**.

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

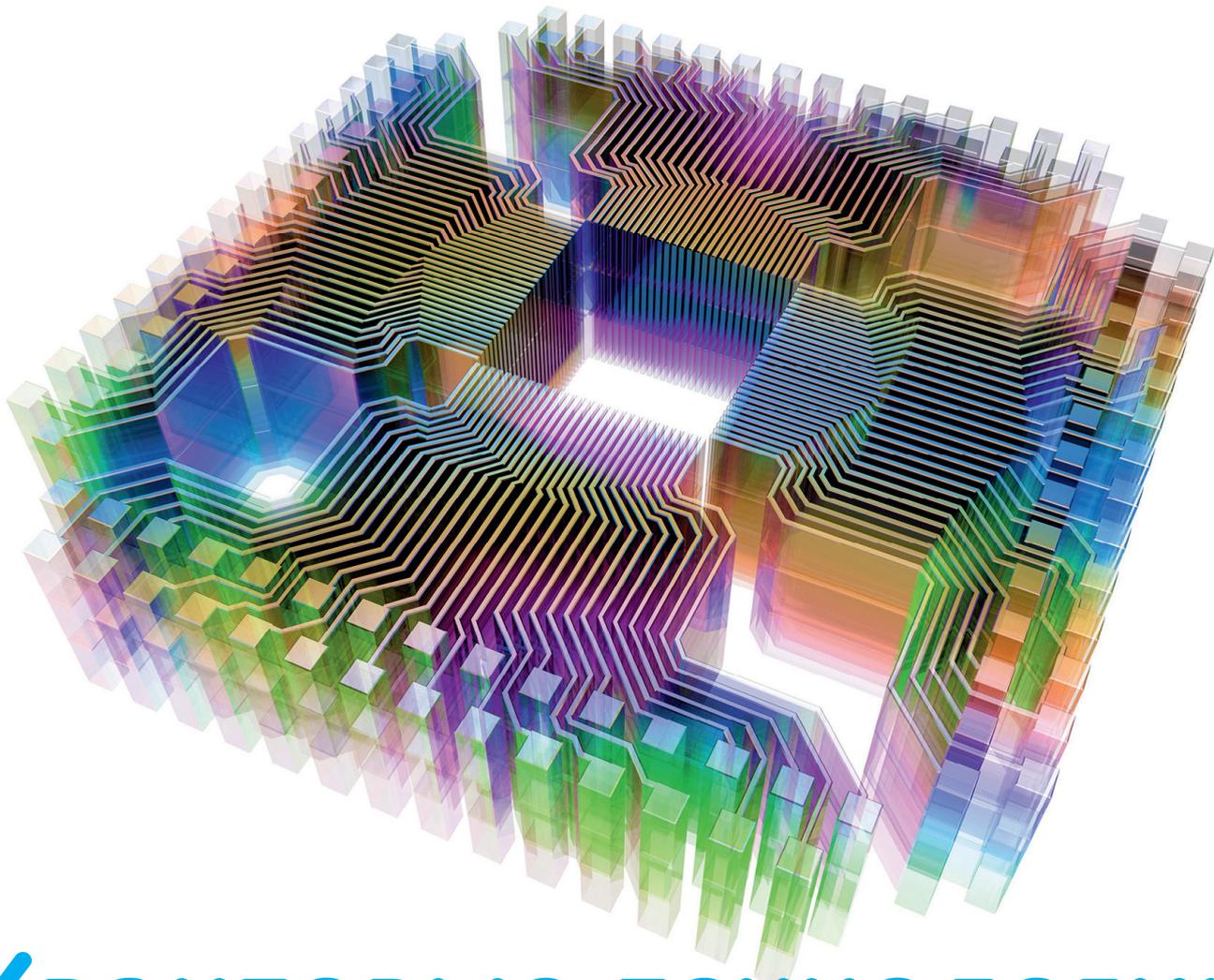
БОНУС

Получение каждого номера в день выхода журнала

* Цена не включает стоимость доставки.

ПОДРОБНОСТИ НА САЙТЕ
WWW.LINUXCENTER.RU/LINUXFORMAT





Квантовые технологии

Матс-Теге Аксельссон вводит вас в мир технологии квантовых компьютеров, самой крутой из всех. Узнайте, как она работает и как ее применять.

Квантовые компьютерные технологии привлекают внимание крупных компаний, ученых и любителей. Наша статья опишет их историю, разные способы создания квантового компьютера и логику программирования. Вы также узнаете о ряде наборов инструментов программирования, которые можно использовать, чтобы приступить к работе.

Для работы с квантовым компьютером надо понимать физические законы, чтобы впоследствии программисты могли употребить и оценить конечный результат. Ученые наблюдают квантовые эффекты в фотонах, электронах и изотопах многих элементов. Это означает, что для создания рабочих квантовых компьютерных систем инженеры применяют такие материалы со сверхпроводимостью, как ниобий и алюминий.

Логические вентили [gate] делаются на кремниевых подложках и управляются СВЧ-генераторами. В долгосрочной перспективе эти решения могут быть не самыми лучшими, но сейчас они работают. Чтобы использовать квантовые компьютеры, вам нужна логика, которая использует две основных концепции:

квантовая суперпозиция и квантовая запутанность [entanglement]. Когда вы начинаете исследовать эти концепции, сфера Блоха [Bloch] помогает вам визуализировать действия с разными вентилями. Для создания требуемых алгоритмов программисты вместе с квантовыми логическими вентилями могут использовать классические двоичные логические вентили.

Популярные медиа восхваляют квантовые компьютеры как намного более быстрые по сравнению с текущими моделями. Но они оказываются быстрыми только в определенных областях: криптографии, оптимизации, моделировании и поиске по базам данных. В криптографии многие алгоритмы безопасны, потому что факторизация больших простых чисел на классическом компьютере займет слишком много времени с практической точки зрения. Квантовый алгоритм Шора может сделать это за считанные минуты. Оптимизация и моделирование могут выиграть от способности квантового компьютера тестировать много решений одновременно.

Поиск по базе данных быстрее в четыре раза. А с более быстрым поиском по базе данных машинное обучение тоже намного ускоряется.

История квантовых компьютеров началась с квантовой физики. В 1900 г. Макс Планк [Max Planck] первым предположил, что свет перемещается дискретными порциями, названными им «кванты». Позднее это привело Эйнштейна [Einstein] к доказательству, что Планк был прав. При измерении фотоэлектрического эффекта ученые могут наблюдать групповое поведение. Эти два открытия позднее породили квантовую физику. Однако чем глубже ученые погружались в квантовую природу вещей, тем сложнее становилось объяснить, как она работает. Для квантовых компьютерных технологий самыми интересными достижениями являются запутанность и квантовая суперпозиция.

Квантовая суперпозиция — это явление, при котором частица существует во многих положениях одновременно. Рассматривая его, ученые заключили, что можно создать квантовый компьютер. Квантовый компьютер — это тот, который выполняет множество вычислений за одну операцию.

Кот в ящике

Знаменитый физик Шрёдингер [Schrödinger] придумал мысленный эксперимент. В нем он описывает кота и газовый контейнер в ящике. Газ отравит кота после радиоактивного распада атома, испускающего газ. Процесс случаен, поэтому наблюдатель не знает, жив кот или мертв, пока не откроет ящик. В квантовой физике это означает, что кот одновременно и жив, и мертв, пока кто-то не посмотрит на него. В случае кота это абсурдно, но для квантовой физики — вполне нормально. Суть высказывания Шрёдингера в том, что должно быть еще некое объяснение, пока что остающееся неоткрытым.

Другое явление — запутанность, где две частицы могут иметь переплетенные состояния [intertwined states]. Это означает, что состояние одной частицы всегда будет противоположно другой, даже когда они разделены. В квантовом компьютере запутанность создается программами: это состояние создает логический вентиль CNOT.

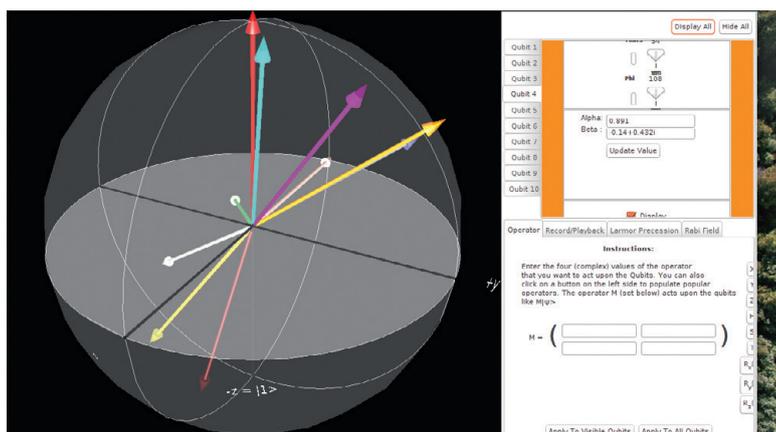
Чтобы использовать все феномены квантовой физики, ученым нужен способ описать происходящее в микромире. Чтобы запрограммировать компьютер, нужны логические операции, описываемые через логические вентили. Квантовые логические вентили и классические логические вентили являются одинаковыми только до определенного момента. Квантовые логические вентили добавляют функции для изменения состояния и запутанности.

Первые попытки создать теорию применения квантовых эффектов в компьютерах начались только в 1970-х. Теория информации Шеннона [Shannon] описывает классические логические вентили и прочие аспекты обработки данных. Для квантовых компьютеров этого недостаточно, поскольку именно квантовых эффектов эта теория не описывает.

Впервые попытка создания квантовой информационной теории была предпринята в 1976 г. В течение 1980-х ученые продвинулись дальше, частично благодаря конференциям по квантовым компьютерным технологиям, организованным MIT и IBM. Среди других интересных разработок были квантовая криптография и первый универсальный квантовый компьютер.

Чтобы использовать все состояния измеряемой частицы, программистам нужен формальный язык. Квантовая информационная теория должна была совершенствоваться. Разные логические вентили являются ключами к подобной схеме.

Отслеживать различные квантовые состояния сложно и запутанно. Прежде всего, то, как это работает, в лучшем случае контринтуитивно. Во-вторых, есть множество разных осей вращения, которые надо отследить. Система несложная, но она предполагает атипичные подходы, которые трудно ухватить. Чтобы понять все переходы состояний, физики предложили сферу Блоха.



Программисты используют сферу Блоха для иллюстрации того, как квантовые логические вентили работают с кубитами.

В 1994 г. Питер Шор [Peter Shor] разработал алгоритм, названный его именем. Этот алгоритм решает проблему обнаружения простых множителей больших чисел. Основой всей криптографии является то, что эту проблему невозможно решить за некое разум-

ШРЁДИНГЕР ЛЮБИЛ КОТОВ!

«Процесс случаен, поэтому наблюдатель не знает, жив кот или мертв, пока не откроет ящик. В квантовой физике это означает, что кот одновременно и жив, и мертв.»

ное время. Квантовые компьютеры способны решить эту проблему за минуты. Интерес к квантовым компьютерам взлетел до небес.

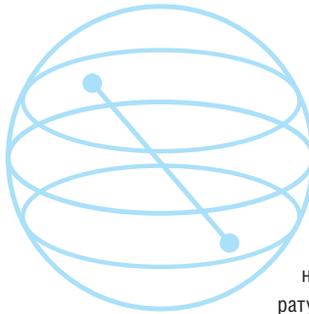
Вскоре после открытия Шора Лов Гровер [Lov Grover] изобрел алгоритм поиска по базе данных, полезный и для других задач (также названный именем автора).

» ШИФРОВАНИЕ ПОД УГРОЗОЙ

Большой вопрос заключается в том, создают ли квантовые компьютеры риск для систем шифрования. Основную озабоченность вызывает схема шифрования RSA. Ее безопасность зиждется на том условии, что разложение большого числа на простые множители занимает слишком много времени. Для попыток отыскать простые числа есть множество стратегий, и самой простой будет метод угадывания. Метод проб и ошибок непрактичен, поскольку для 2048-битного числа имеются миллионы решений. Некоторые стратегии могут сократить число возможных решений, но даже на самые эффективные методы уйдут годы или миллионы лет. С правильным алгоритмом квантовый компьютер может сократить это время до вполне практического уровня. Предпринимаются попытки создать другие алгоритмы, которые нельзя будет взломать таким образом.

Это благоразумное начинание, но риск того, что квантовые компьютеры смогут сделать это в ближайшие 15 лет, не высок. Имеющиеся сегодня квантовые компьютеры небольшие, и их трудно программировать. Доступных для программирования сред мало, и они редки. Как вы увидите из других частей нашей статьи, вы всё же можете настроить несколько состояний кубитов и переплести состояния. Превращение этого в тонко настроенную технологию взлома шифрования, вероятнее всего, является очень отдаленной перспективой.

В следующие десятилетия ученые сделали новые открытия. Главными были исправление квантовых ошибок и отказоустойчивые квантовые компьютеры. Первая демонстрация квантового компьютера состоялась в 1998 г. После этого их развитие ускорилось.



по-прежнему сложно. Кроме того, производство с применением современных технологий весьма дорогостоящее. Между прочим, конденсат Бозе–Эйнштейна — это газ, охлажденный до температуры в милликельвинах [т.е. близкой к абсолютному нулю, — прим. ред.]. Когда газ достигает этой температуры, все электроны принимают одинаковое квантовое состояние. Использование этого решения пока что ограничивается симуляторами — настоящие компьютеры остаются несбыточной мечтой.

Заходят в бар Бозе, Эйнштейн и Джозеф...

Ныне существует целый ряд работающих квантовых компьютеров. Исследователи используют ионные ловушки, световые и управляемые СВЧ конденсаты Бозе–Эйнштейна. Им нужны разные механизмы управления. Ионные ловушки используют электромагнитные поля для улавливания ионов. Лазеры создают поле, удерживают уловленный ион и «накачивают» состояние электрона в ионе. Электрон будет испускать свет только когда состояние будет соответствовать частоте лазера.

Это логическое состояние 1, а логическое состояние 0 противоположно ему. Это упрощает измерения, но добиться точности

Интересны различные типы с джозефсоновским [Josephson] контактом — именно они работают в реальных устройствах. Версии IBM даже доступны для всех, кто зарегистрировался на странице Q-experience.

Джозефсоновские контакты использует D-волновой компьютер. Это сверхпроводящие логические вентили из ниобия и оксида алюминия. При температурах около абсолютного нуля SQUID измеряет их электромагнитные свойства. SQUID — это чувствительный магнитометр. Значения показывают квантовое состояние контакта, превращая цепь в кубит. Принцип, используемый этим компьютером — квантовый отжиг.

Решение IBM — ядерный магнитный резонанс (Nuclear Magnetic Resonance, NMR). Кубит — по-прежнему джозефсоновский вентиль. Это решение для взаимодействия с кубитом использует СВЧ-передатчики.

Разница между ними в том, что решение IBM в большей степени контролирует кубиты. И это облегчило канадской компании D-Wave производство более крупных чипов. В то же время это ограничивает области применения D-волн. Например, система на основе отжига не может запускать алгоритм Шона или Гровера. Она удобна для оптимизации и машинного обучения — неудивительно, что у Google уже есть один из квантовых D-wave компьютеров.

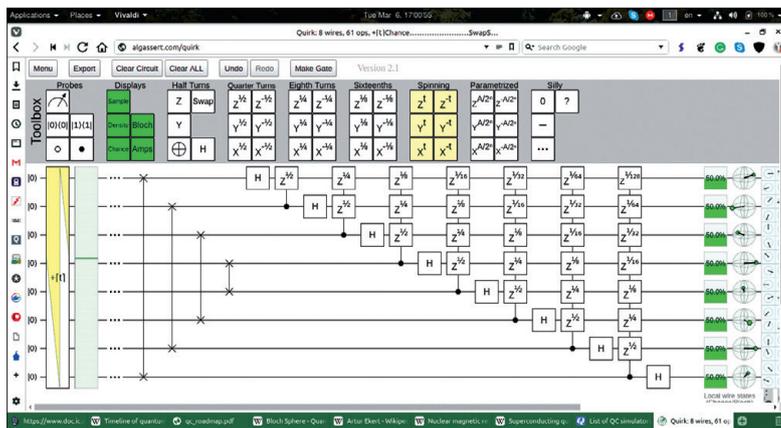
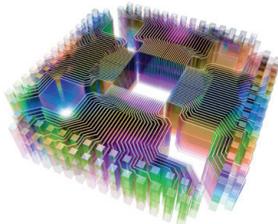
Чем больше, тем меньше работы

Как уже упоминалось ранее, алгоритмы в квантовых компьютерах разные. Чтобы объяснить разницу, вы должны знать используемые законы физики. Как известно большинству читателей *Linux Format*, обычные компьютеры используют логические вентили. Система использует Булеву логику для создания алгоритмов вычислений. Самый важный ее аспект в том, что она детерминированная, и мы имеем дело только с определенностью. Ответом будет или да, или нет — по крайней мере, на уровне электросхемы.

В противоположность ей, квантовые вычисления включают все ответы, имеющие ненулевую вероятность. Ни один ответ не является точным — скорее, он вероятностный. Чтобы это было эффективным, нужно соответственно адаптировать алгоритмы.

Квантовые логические вентили воздействуют на один или несколько кубитов одновременно. При измерении кубитов система обычно делает полный анализ всей системы полностью. И если вы ищете один кубит, в одной операции тот будет выделен, как нарыв на пальце. Обратите внимание, что одна операция состоит из множества запусков одних и тех же логических вентилях. Это одна из причин того, почему квантовые компьютеры быстрее обычных.

Самый общий сценарий — найти особое значение из множества, например, сделать поиск по базе данных. Другой часто упоминаемый случай — взлом шифра. С помощью алгоритма Шора вы измеряете решение действием логических вентилях. Логические вентили запускают введенную информацию определенное число раз, прежде чем найти ответ. Это число не меняется, если введенное число больше. Классический компьютер заставляет вас проверять все возможные ответы. Это означает,



В самом начале программисты используют партитуру, как в музыке. На рисунке показан Quirk, онлайн-симулятор. Схема изображает квантовое преобразование Фурье.

» КВАНТОВОЕ МОДЕЛИРОВАНИЕ

В помощь пониманию, как работают квантовые компьютеры, придумано немало симуляторов. Эффективную симуляцию вы сделать не сможете, но если вы хотите понять внутренние принципы, то многих из имеющихся будет вполне достаточно. Первый из достойных исследований — симулятор сферы Блоха. Без хотя бы рудиментарного представления о том, что такое сфера Блоха, вам будет очень трудно понять, что делают логические вентили. Вы можете узнать об этом больше на <http://eecs.ceas.uc.edu/~cahaymm/blochsphere>.

Есть и другие симуляторы — большинство из них является результатом докторских диссертаций и по этой причине не являются действующими: это просто неплохое начало. Некоторые из упомянутых здесь будут полезны для самообразования.

Симулятор JQuantum выглядит и ведет себя как квантовая партитура. Он немного громоздок в использовании, но если вы попрактикуетесь, читая учебник, станет вполне пригодным к использованию и понятным. Также он поддерживает пользовательские функции.

Это очень удобно, когда вы учитесь и хотите проверить это самостоятельно. Единственный другой способ сделать это — использовать свою IDE и самостоятельно написать квантовую программу. Если у вас есть опыт работы в Python, такое вполне можно сделать, но должны предупредить вас — красивой симуляция не будет!

что когда значение достигнет 2048, вам, вероятно, потребуется миллион лет, чтобы его решить. Больше чисел означают экспоненциальный рост нагрузки. Обратите внимание, что в квантовом компьютере усилия на решение задачи не увеличиваются.

Устроим вечеринку Блоха

Узнав, сколько состояний и переходов в каждом кубите, пора браться за код. Однако не стоит сильно спешить — есть визуальный способ описать логические вентили. Сфера Блоха — основа для всех логических вентилях. График похож на нотную запись, поэтому называется квантовой партитурой. Они немного разные для разных платформ, но в целом поперечные струны — это кубиты. Желая изменить состояние кубита, вы помещаете на струну логический вентиль. Чтобы создать сцепление двух кубитов, вы помещаете логический вентиль на одну струну и соединяете его с другой.

Для описания квантового состояния частицы вы используете вращение $[\text{spin}]$, как у закрученного мяча. Точнее, угол вращения «мяча». Хотя в классическом смысле никакой мяч не вращается, математика хорошо это описывает. Когда оборудование замеряет электрон, его спин будет направлен вверх или вниз по оси. Система может делать оценку в одном измерении за один раз. Три — это x , y и z . И если вы измеряете x , вы теряете всю информацию об осях z и y .

Описывая квантовое состояние, вы используете воображаемую стрелку в сфере Блоха.

Верх сферы — это логический 0 кубита, а низ — логическая 1. Существует только два состояния, которые система может измерить с определенностью 100%. Чтобы получить результат в квантовом компьютере, вы делаете множество «проб» и получаете наиболее вероятный ответ.

Матрица содержит координаты, которые показывают угол стрелки. Длина стрелки всегда равна 1. Значения, которые мы используем для вычисления вероятности, описывают компоненты оси стрелки. Сфера Блоха управляет логическими вентилями и их именованиями, так что помните об этом, когда будете пытаться понять.

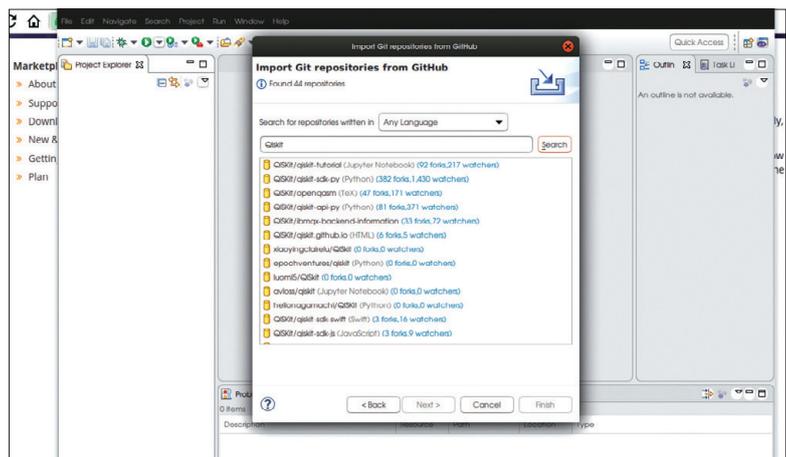
Имеющиеся логические вентили зависят от приложения и типа квантового компьютера. Но наиболее общими являются следующие. X-вентиль Паули [Pauli] поворачивает состояние по оси x , меняя значение Z . Этот логический вентиль также является логическим вентилем инвертирования разрядов. Y-вентиль Паули вращается по оси y , изменяя значения x и z . Это логический вентиль инвертирования разрядов и переброса фазы.

Пока что логические вентили только переключают биты и фазы. Это не имеет отношения к суперпозиции и запутанности. Вентиль Адамара [Hadamard] помещает кубит в суперпозицию. Поясняем: у кубита есть 50-процентный шанс стать 0 или 1, переключаясь между X и Z.

Это похоже на игральные кости, только дубль всегда дает начальное значение. Другие логические вентили, которые расширяют вентиль Адамара — S и его преобразованный сопряженный вентиль.

Чтобы использовать запутанность, вы используете логический вентиль CNOT, обычно отображаемый со знаком плюс в круге и линией вниз к другому кубиту. Логический вентиль переключает целевой кубит, когда регулятор равен 1.

Это основные логические вентили, с которыми вы встретитесь, когда начнете работу. В Q Experience от IBM у вас есть возможность создавать подпроцедуры. Однако лучше будет использовать инструмент разработки. Их доступно много, и вы можете с их помощью расширить свою IDE. В Eclipse, Netbeans и Visual Studio имеются модули, использующие некоторые классические языки. Например, инструменты программирования



Здесь мы используем Eclipse для разработки наших собственных квантовых компьютерных программ. Обратите внимание, что данный конкретный инструментальный поддерживается языком программирования Python.

ПОСТАВЬТЕ НА МЕНЯ

«В противоположность ей, квантовые вычисления включают все ответы, имеющие ненулевую вероятность. Ни один ответ не является точным — скорее, он вероятностный.»

на Python доступны от IBM. Обратите внимание, что процедуры готовят скрипты для квантового компьютера — на нем не запускается Python.

Итак, приступайте. Сначала вы создадите не очень много, но получите массу удовольствия — это мы вам гарантируем! **LXF**

» ИНСТРУМЕНТЫ ОТ УМНИКОВ

Самостоятельное программирование требует стандарта под названием OpenQASM. Это основа для всех наборов разработки.

Многие группы разработали инструментари с помощью этого стандарта. Наиболее известные — инструментари от IBM, D-Wave и Microsoft.

IBM решили использовать Python для создания Qiskit, и вы можете скачать этот инструментальный с Github. У него также множество исходников и демонстрационных наборов. Здесь вы можете узнать всё о текущем состоянии разработки.

И в Eclipse, и в Netbeans всё, что вам надо сделать — это импортировать код в проект и исследовать его. Не забудьте до начала компиляции установить Python 3.5 или выше. IBM QE имеет эти примеры в своем инструментари Python.

Установив исходники, вы можете только запустить симуляцию на своем компьютере. Если вы хотите попробовать реальный квантовый компьютер, зарегистрируйте учетную запись на IBM Q Experience. Все очень просто: вам надо только открыть учетную запись и получить ключ API с вашей учетной записи. Затем вставьте его в файл `Qconfig.py` проекта, над которым вы работаете. Для использования реальных компьютеров имеется система кредитования, поэтому делайте симуляцию, пока не убедитесь, что вы всё делаете правильно. Если вы действительно умны, то сможете достичь экспертного уровня. Тогда вы получите больше устройств для проведения экспериментов.

Если вы используете Visual Studio, то сможете скачать файл `Qvis` и добавить расширение. Но вам всё равно нужна поддержка Python.



Devuan: плод нелюбви к systemd

Евгений Гольшев копается в причинах появления Devuan, одного из ярчайших представителей семейства Debian'овых.

Кена Томпсона [Ken Thompson] однажды спросили, что бы он изменил в UNIX, если бы проектировал систему заново, и он ответил: «Я бы написал `creat` с буквой *e*» (англ. *I'd spell creat with an e*). Эта шутка, на мой взгляд, как ничто другое подчеркивает, насколько бережно Unix хранит традиции: за 40 с лишним лет так никто и не добавил букву *e* к имени системного вызова, хотя количество опечаток, допущенных при написании кода, велико и с каждым днем продолжает расти. В конце концов, кто может осмелиться сказать, что архитектурные решения патриархов Томпсона и Ритчи ошибочны?

Unix — это удивительный мир, в котором желание к познанию чего-то нового уживается со слепой верой и следованию традициям.

И однажды в этот мир вторгся человек по имени Леннарт Пёттеринг [Lennart Poettering], со своим видением дистрибутивостроения, которое, по мистическому совпадению, в корне отличалось от того, к которому все успели привыкнуть за пару десятков лет. Всё бы ничего, но Леннарт родился с редкой способностью изменять всё вокруг, поэтому он легко начал воплощать свои задумки, а через крупнейшие дистрибутивы (Debian, Ubuntu, Red Hat Enterprise Linux, SUSE Linux Enterprise и т. д.) они стали добираться до согласных и несогласных с ним пользователей.

Данная статья посвящена Devuan (читается как DevOne). Devuan является производным от Debian GNU/Linux дистрибутивом, основная цель которого заключается в альтернативном развитии Debian, как если бы он никогда не встретился с известным детищем Леннарта под названием `systemd`. Я написал эту статью главным образом по двум причинам.

Во-первых, Debian-подобные дистрибутивы — моя страсть, а Devuan, один из ярких представителей этого семейства, упоминался на страницах **LXF** только вскользь, поэтому я чувствую своим долгом исправить это недоразумение.

Во-вторых, я хочу помочь читателям разобраться в ситуации, сложившейся вокруг проекта `systemd`, а также понять, что побудило разработчиков Devuan потратить два с половиной года на то, чтобы «вытравить» `systemd` из Debian Jessie, получив таким образом первую стабильную версию производного дистрибутива.

В конце концов, если человек в один прекрасный момент поймет, что ненавидит `systemd`, то пусть хотя бы будет способен объяснить, почему.

К счастью или сожалению, материал не содержит информации, которая могла бы оказаться полезной при

- установке, настройке и администрировании серверов под управлением Devuan;
- администрировании серверов на базе Linux-подобных операционных систем, использующих `systemd` (если вы интересуетесь практической стороной `systemd`, то рекомендую обратиться к **LXF199**).

Хочу сразу отметить, что я не являюсь пользователем Devuan, предпочитая ему Debian (рабочая станция) и Ubuntu (ноутбук), и стараюсь занимать нейтральную позицию в спорах, связанных с действиями Леннарта Пёттеринга в общем и с `systemd` в частности. Но разочаровываться не стоит — я регулярно имею дело с Devuan при решении различных задач, одной из которых является разработка сервиса для кастомизации образов операционных систем для одноплатных компьютеров.



Ленарт полон неоднозначных идей того, каким образом дистрибутивы GNU/Linux можно сделать лучше.

А теперь предлагаю начать обсуждение заявленных тем... но обо всём по порядку.

В погоне за ускорением загрузки

В конце 2000-х несколько крупных производителей операционных систем на базе ядра Linux пришли к выводу, что необходимо ускорять время загрузки своих систем. Так как за загрузку операционной системы отвечает система инициализации, стало очевидно, что начинать нужно именно с нее. В течении долгого времени дистрибутивы GNU/Linux использовали для этих целей SysVinit, созданную Микéлем ван Смооренбургом [Miquel van Smoorenburg] и основанную на идеях оригинальной System V init. SysVinit сейчас считается классической системой инициализации и в данной статье противопоставляется всем другим.

SysVinit представляет собой систему инициализации, *основанную на уровнях запуска* (runlevel-based). Уровни запуска — это состояния, в которых может находиться система; система переходит из одного состояния в другое. SysVinit выполняет скрипты, ассоциированные с определенным уровнем, когда система входит в (или покидает) этот уровень. К примеру, когда система входит в уровень запуска N, то SysVinit запускает скрипты из /etc/rcN.d, где N — число от 0 до 6. Однако со временем накопилось много потребностей, которые классическая система инициализации не могла удовлетворить в силу своей архитектуры (к примеру, слежение за своими дочерними процессами и их перезапуск в случае необходимости).

Недостатки SysVinit мотивировали компанию Canonical, которая является основным разработчиком Ubuntu, начать работу над альтернативной системой инициализации под названием Upstart, а достоинства launchd из Mac OS X и SMF из Solaris послужили основным источником ценных идей, которые помогли в решении поставленной задачи. Новая система инициализации оказалась эффективнее, чем SysVinit благодаря *событийно-ориентированной архитектуре* (event-driven architecture). Системные сервисы стало возможным запускать и останавливать на основе

событий, а не уровней запуска. Данный подход предполагал генерацию событий при запуске или остановке системных сервисов, что позволило организовать привязку к ним других сервисов. Таким образом, для определения последовательности запуска

» ИМЕНА ДОВЕРЬТЕ АВТОРАМ

Есть иногда у некоторых проектов такие имена, глядя на которые, не сразу можно угадать их правильное произношение (под «правильным» я в данном случае подразумеваю принятое в сообществе). Обычно в таких случаях необходимо обратиться к FAQ'у или истории этих проектов. В качестве примера на ум приходят web-сервер Lighttpd (читается как lighty) и Nginx (читается как engine-x). (Очень забавно, когда люди с серьезным выражением лица рассказывают о некоем web-сервере под названием Энгинкс.) Но история с именами на этом не заканчивается.

Есть и такие проекты, авторы которых настаивают на строгом написании названий своих детищ. Однако все мы привыкли к тому, что слова, с которых начинаются предложения, пишутся с большой буквы. А еще мы помним, что сглавной буквы всегда пишутся все имена собственные. Но в школе обычно не учат тому, что всем этим правилам нужно следовать только в том случае, если они не противоречат правилу написания конкретного имени. В качестве примеров проектов, с которыми нужно считаться с этой точки зрения, хочу привести ownCloud и, конечно, *systemd*. Таким образом, в каком бы месте предложения ни стояли эти имена, после какого бы знака препинания не шли — они должны сохранять авторский вариант написания.

В заключение, думаю, что нужно сказать пару слов о том, почему следует писать *systemd* с маленькой буквы несмотря на то, что это имя собственное. Дело в том, что проект был назван по имени основного исполняемого файла, который запускается первым при загрузке системы, становясь процессом с номером 1 и родителем всех остальных процессов. Традиционно (и тут тоже традиции!) в UNIX и C, родного для этой ОС языка, использовался нижний регистр, поэтому одна из ключевых программ в Linux-подобной операционной системе могла быть названа только *systemd*, а не *Systemd*, *SystemD* или как-то еще.

Второй в истории лидер проекта Debian является пользователем Devuan.



сервисов и оценки возможности их выполнения в параллельном режиме был взят на вооружение метод учета зависимостей. Работа в тандеме с ядром, которое также стало развиваться в сторону событийно-ориентированной архитектуры, благодаря чему появилась возможность асинхронной загрузки драйверов, заметно сократила время от нажатия кнопки включения до полной готовности всей системы. И при всём при этом Upstart была разработана с оглядкой на обратную совместимость с SysVinit.

Upstart дебютировала в Ubuntu 6.10 “Edgy Eft” в конце 2006 г., заменив SysVinit (однако окончательно новая система инициализации была интегрирована в дистрибутив только к выходу Ubuntu 9.10 “Karmic Koala”). Затем на Upstart обратили внимание компании Google и Red Hat, интегрировав ее в свои продукты. Так Upstart нашла применение за пределами Ubuntu и начала использоваться в ChromeOS и Red Hat Enterprise Linux 6. На новую систему инициализации даже планировал перейти Debian.

При всех своих достоинствах, Upstart оказалась всего лишь трамплином для другой, более совершенной системы инициализации. Так, в апреле 2010 г. компания Red Hat, при участии разработчиков из Novell, IBM, Intel и Nokia, поставила перед собой цель создать систему инициализации, нацеленную на более интенсивную параллелизацию выполнения сервисов на этапе загрузки. Проект, получивший название *systemd*, возглавил Леннарт Пёттеринг. И уже в июле того же года была выпущена первая стабильная версия проекта.

На тот момент *systemd* получала не больше негативной критики, чем любой другой проект, позиционирующий себя как альтернативная система инициализации. Всё шло относительно

гладко, несмотря на то что *systemd* (почти) сразу обратила на себя внимание своим агрессивным подходом к делам, связанным с обратной совместимостью. (К примеру, если Upstart была совместима со скриптами инициализации SysVinit, то новейшая система инициализации не была завязана ни на *Bash*, ни на любую другую оболочку.) Ситуацию не сильно усугубил и тот факт, что именно Леннарт был *тем самым* разработчиком звукового сервера *PulseAudio*, который в начале своего пути не отличался стабильностью работы (в основном из-за аудиодрайверов, а не ошибок в коде *PulseAudio*).

Проекту серьезно досталось немного позднее, когда в один прекрасный день *systemd* перестал позиционироваться как альтернативная система инициализации, превратившись в системный менеджер, в котором система инициализации стала всего лишь одним из многочисленных компонентов. О двух таких компонентах — *journal* и *logind* — я хочу рассказать прямо сейчас.

logind представляет собой систему управления пользовательскими сеансами. До того, как за решение этой задачи взялись разработчики *systemd*, дистрибутивы GNU/Linux для этих целей использовали *ConsoleKit*. В свое время *ConsoleKit* поставил перед собой амбициозную цель — поддержку *более одного независимого рабочего места* (*multi-seat*), что предполагает одновременную работу на одной системе нескольких независимых графических сеансов для различных пользователей. (Функция редкая, но в некоторых случаях очень востребованная.) Тем не менее, архитектура *ConsoleKit* не способствовала достижению заветной цели и проект оказался в глубокой стагнации. Так, без переизобретения велосипеда непросто задача поддержки более одного

независимого рабочего места была решена в рамках компонента logind. Другими словами, лучшей альтернативы в мире Linux еще не видели. Более того, решение оказалось настолько удачным, что Энди Винго (Andy Wingo), один из разработчиков дистрибутива Guix, реализовал возможность обособленного использования logind. Проект получил название elogind (по аналогии с eudev). На elogind даже перешел Devuan, который, напомним, принципиально не использует *systemd*; но об этом немного позже.

journald представляет собой сервис журналирования событий. Этот компонент, в свою очередь, является отличным примером того, как разработчики *systemd* не только решили задачу, которая давно не нуждалась в решении, но и решили ее по-своему. Двумя основными отличиями journald от различных реализация syslogd, которые много лет занимались задачами журналирования в Unix, являются использование криптографических средств для гарантирования неизменности и целостности накопленных логов и хранение этих логов в двоичном виде. Традиционно логи на Unix-серверах хранятся в текстовом виде, поэтому хранение логов в двоичном виде воспринимается как недостаток, который обесценивает любые достоинства journald, даже такие крутые, как гарантию неизменности и целостности логов. Подробнее о journald можно узнать из **LXF191/192** и **LXF199**.

Появление Devuan

В конце 2013-го — начале 2014-го Debian в очередной раз вернулся к актуальной для себя задаче — переходу на более современную систему инициализации. В качестве кандидатов рассматривались Upstart и *systemd*, и по результатам голосования победила последняя. Следом за Debian пошла Ubuntu.

По итогам интеграции *systemd* в Debian дистрибутив не только перешел на новую систему инициализации, но и оказался сильно завязанным на системном менеджере, который был призван управлять всеми аспектами работы системы. В итоге, *systemd* был поставлен в один ряд с ядром и стандартной библиотекой языка C, для которых в дистрибутиве не предусмотрены альтернативы. Jessie стал первым выпуском Debian, который не мог функционировать без *systemd*. Таким образом, группа разработчиков Veteran Unix Admins, которые были не согласны с этой политикой, выпустили в рамках проекта Devuan свободную от *systemd* версию Jessie. Чтобы этого добиться, разработчикам потребовалось внести изменение в 381 пакет. Так появился первый выпуск Devuan, позволивший пользователям Debian использовать или не использовать *systemd* в зависимости от их желания.

В качестве системы инициализации в Devuan по умолчанию используется SysVinit, которая сейчас более-менее активно развивается. Дело в том, что версия 2.89, выпущенная 28-го марта 2018-го года, была первой за 8 лет стагнации проекта, но с приходом Джесса Смита [Jesse Smith], который занял пустующее кресло сопровождающего проекта, ситуация резко изменилась.

Различные новостные ресурсы, освещающие достижения открытого и свободного программного обеспечения, позиционируют Devuan как борца за независимость от *systemd* (отчасти это объясняется тем, что самыми оперативными оказываются информационные источники, пополняемые сообществом, а профессиональные издания, как правило, чуть медленнее реагируют на события и потому постоянно догоняют первых по горячим следам). С одной стороны, это позволило быстро обратить на проект внимание, сделав Devuan звездой мира СПО. С другой стороны, это искажает оригинальные цели проекта — свободе выбора системы инициализации. Сейчас с полной уверенностью можно утверждать, что Devuan добился этой цели: во втором выпуске дистрибутива, известном под кодовым именем ASCII, который



Devuan не старается бойкотировать *systemd*, а только отстаивает право выбора системы инициализации.

вышел 9 июня 2018 г., пользователям помимо SysVinit предлагается OpenRC, которая используется в Alpine Linux и Gentoo.

Заключение

Поляризация взглядов в сообществе пользователей и разработчиков свободного и открытого программного обеспечения является обычным делом. Одним из старейших примеров этого явления является вражда пользователей KDE с приверженцами Gnome. Другими словами, людям нужно что-то ненавидеть.

Что касается фактов, то *systemd* является образцом отлично написанного системного программного обеспечения, а в сообществе разработчиков царят мир, гармония и взаимоуважение. Леннарт, в свою очередь, является примером отличного лидера,

О РЕШЕНИЯХ SYSTEMD

«Некоторые компоненты *systemd* уже используются в дистрибутивах, которые сознательно отказываются от *systemd* как от системы инициализации.»

который, ко всему прочему, еще и невероятно креативен — пишет много и интересно, освещая различные особенности Unix в своем блоге. Но чуть подсказывает, что где-то здесь должен быть подвох. И он действительно есть. Когда в рамках *systemd* решаются какие-то задачи, то разработчики, как правило, подходят к их решению по-своему и без оглядки на обратную совместимость, посягая на святая святых — традиции Unix (к примеру, я почувствовал легкое недоумение, когда первый раз прочитал новость о том, что journald предлагает хранить логи в двоичном виде). Именно здесь и начинаются конфликты интересов. Но не все решения разработчиков *systemd* так уже плохи. Об этом свидетельствует тот факт, что некоторые компоненты *systemd* уже используются в дистрибутивах, которые сознательно отказываются от *systemd* как от системы инициализации. **LXF**

Итак: чума на оба ваши дома!

Новые ужасы всплывут о Meltdown и Spectre еще до того, как д-р **Валентин Синицын** убедит вас усилить свою безопасность брандмауэрами на nftable.



**НАШ
ЭКСПЕРТ**

Д-р Валентин Синицын — бывший разработчик KDE. Любит порождать облака Linux и писать статьи.

Еще в начале января группа исследователей безопасности сообщила об атаках бокового канала на большинство x86, выпущенных за последние десять лет. Вскоре эти уязвимости были названы Meltdown и Spectre. Это, конечно, относительно старая история, так зачем мы тратим на нее место?

В марте 2018 г. небольшая исследовательская компания из Израиля обнаружила множественные критические уязвимости в последних процессорах AMD Zen. По традиции эти уязвимости получили кодовые имена: Masterkey, Ryzenfall, Fallout и Chi-mega. У них есть несколько разновидностей, и в общем и целом они могут запустить дюжину векторов атак. Чипы AMD EPYC, Ryzen, Ryzen Pro и Ryzen Mobile подвержены воздействию, и это плохая новость для пользователей AMD на настольных ПК, серверах и мобильных устройствах. Возможные негативные последствия варьируются от похищения аутентификационной

информации сети до установки устойчивого вредоносного ПО, которое переживает перезагрузку BIOS. Все уязвимости требуют локального физического доступа, и большинству из них нужно, чтобы злоумышленник сначала приобрел привилегии локального администратора.

Возможно, вы заметили, что мы не даем вам никаких технических подробностей. Причина в том, что на момент написания их нет, и это уже ставит случай AMD в сторону от истории с Meltdown. CTS Labs опубликовали техническую документацию, но она содержит только самые общие описания. Заявлено, что подробности будут обсуждаться с AMD — за день до объявления — и с другой лабораторией безопасности, Trail of Bits, которая подтвердила, что уязвимости существуют. CTS также отступили от принципа «ответственного разглашения», и руководитель компании, Илайя Люк-Зильберман [Iliia Luk-Zilberman], опубликовал открытое письмо, объясняющее, почему, по его мнению, непредоставление производителю преимущества по времени при сохранении подробной информации в секрете является наилучшим подходом для потребителей. Интересно также, что в правовом уведомлении не указано никакой гарантии отсутствия прямых или косвенных экономических интересов в этом деле.

CTS заявляет, что ошибки — результат несоблюдения AMD «фундаментальных принципов безопасности». AMD смогли поделиться информацией о собственном анализе двумя днями позже: <http://bit.ly/LXF237amd> (1). Поскольку ожидается опубликование еще большего объема информации, что можете с этим сделать лично вы? Просто: планируйте собственную безопасность; используйте соответствующие инструменты; и никогда не прекращайте учиться, и для удовольствия, и для безопасности.

» DEVOPS ИЛЬ НЕ DEVOPS?

Вы когда-нибудь слышали о такой вещи, как DevOps? Могу поспорить, что да. Не потому, что это нечто потрясающее или особо скверное. Я вижу это слово в каждом втором списке рассылки, на которые я подписан.

Означает ли это, что в Рецептах больше не надо рассказывать об администрировании? Хороший вопрос. Мир вокруг нас меняется, а ландшафт ИТ меняется особенно быстро. Многие, если не все, должности стали многофункциональными, и почему разработчики и сисадмины должны стать исключением?

Я согласен, что разработка программ и обеспечение их ровной работы на предприятии предполагает наличие немного другого образа мысли. Но не понимаю, почему эти разные образы мысли не могут объединяться в одном человеке. Возьмите системного администратора и научите его определенным шаблонам разработки программ. Возьмите своего разработчика движков и расскажите ему, что означает средняя нагрузка и как работать с *tcpdump*. И тот, и другой — инженеры, поэтому им понравится изучать нечто новое. Конечно, обучение требует времени. Если вы просто объедините свою операционную команду и команду разработки, требуемые навыки не возникнут у них по волшебству. А время — деньги, так что будьте готовы потратиться сейчас, чтобы вернуть эти инвестиции в будущем.

Может показаться, что я защищаю DevOps. На самом деле, нет. Как любая методика, для кого-то она может не сработать. Если вы не рассчитываете на возврат своих инвестиций, не делайте их. А может быть, стоит начать не спеша: пусть ваши разработчики узнают, что и как работает внутри компьютера, и через некоторое время они смогут работать с круглосуточными звонками. Именно такой путь я выбираю в Рецептах. Как вы считаете, он сработает? Напишите мне: valentine.sinitsyn@gmail.com.

Number	Vulnerabilities	System Workstation	System PC	System Mobile	EPYC Server
1	MASTERKEY-1	●	●	●	●
2	MASTERKEY-2	●	●	●	●
3	MASTERKEY-3	●	●	●	●
4	RYZENFALL-1	●	●	●	●
5	RYZENFALL-2	●	●	●	●
6	RYZENFALL-3	●	●	●	●
7	RYZENFALL-4	●	●	●	●
8	FALLOUT-1	●	●	●	●
9	FALLOUT-2	●	●	●	●

Карта уязвимостей [Vulnerabilities Map] на <https://amdflaws.com> выявляет подтвержденные и потенциальные эксплойты в системах AMD.

Nftables: изолируем брандмауэр

В чреватом опасностями современном мире мы хотим надежно оградить периметр своей сети. К счастью, в Linux есть для этого инструменты...

Все мы ценим Linux за его свободу, стабильность и безопасность — но он не становится безопасным по волшебству. Как объяснялось в предыдущей статье (см. **LXF232**), его можно сделать безопасным, употребив правильные инструменты. На практике мы часто рассчитываем на помощь производителя дистрибутива, однако сохраняется наше изначальное наблюдение: инструменты безопасности — краеугольный камень вашей системы. И ваш брандмауэр — хороший тому пример.

Настольные ПК или серверы в небольшом офисе часто работают практически без явно настроенных правил брандмауэра. Это нормально, потому что такие системы обычно не используют сложной политики управления доступом. В сети предприятия всё иначе: там вам надо, чтобы доступ к базе данных по персоналу был у бухгалтеров, но не у сотрудников отдела продаж.

Когда речь заходит о брандмауэре, опытный администратор Linux сразу вспоминает об *iptables*. Они служили нам верой и правдой почти двадцать лет, но теперь есть более изящная альтернатива: *nftables*.

Фильтры в ядре

В общих чертах, лучше всего избавляться от ненужных пакетов как можно раньше. Однако если вы боретесь с DDoS-атакой, вполне нормально будет избавляться от пакетов по мере их продвижения по сетевому стеку Linux. Для этих целей ядро реализует т.н. Netfilter hooks [ловушки сетевых фильтров]:

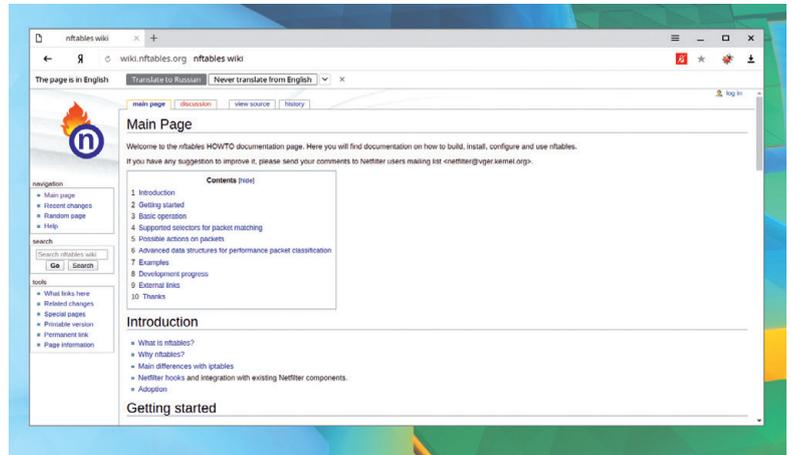
```
int ip_local_deliver(struct sk_buff *skb)
{
    ...
    return NF_HOOK(NFPROTO_IPV4, NF_INET_LOCAL_IN,
        net, NULL, skb, skb->dev, NULL,
        ip_local_deliver_finish);
}
```

Функция ядра *ip_local_deliver()* заново собирает фрагменты IP-адреса (что в данный момент не существенно) и доставляет пакет в программу работы с протоколом более высокого уровня: например, TCP. Большая часть этого на самом деле происходит в функции *ip_local_deliver_finish()*, которая вызывается только если все ловушки, зарегистрированные для протокола IPv4 на номере *NF_INET_LOCAL_IN*, возвращают успешный результат (*NF_ACCEPT* или *NF_STOP*).

К одной ловушке может быть прикреплено несколько функций. Регистрируя функцию ловушки, вы также указываете значение приоритетности. Пункты с более низким (часто отрицательным) приоритетом вызываются первыми.

В Linux 4.2 *nftables* представили новую ловушку — *NF_NETDEV_INGRESS*. Следуя классической мантре «чем раньше, тем лучше», она запускается, как только пакет входит в сетевой стек Linux. Это происходит сразу после того, как у канала пакета (например, *tcpdump*) будет возможность проверить пакет, что, таким образом, приближает *nftables* к *tc* или даже XDP (более подробную информацию см. в **LXF231**).

Хотя брандмауэры (в прямом смысле) привязаны к ядру, было бы непрактично программировать новый модуль ядра для каждого набора правил. Поэтому и *nftables*, и *iptables* создают общую инфраструктуру, чтобы интерпретировать правила, которые вы пишете на высокоуровневом языке области пользователя.



Разница между ними в том, как реализуются соответствующие виртуальные машины и какой именно это высокоуровневый язык.

Таблицы и цепочки

С точки зрения пользовательской области *nftables* состоит из команды *nft* и вспомогательной библиотеки *libnftnl*. Часть ядра обычно поставляется в виде модулей (*nf_tables.ko* со товарищи), если только вы не используете действительно старый дистрибутив Linux. Если нет, то и *libnft*, и *nft* должны также быть доступны в репозиториях вашего менеджера пакетов. Чем новее ваш дистрибутив, тем приятнее будет работа с *nftables*.

Как и *iptables*, *nftables* сначала подключают правила брандмауэра, затем группируют их в таблицы (отсюда и название). Однако, в отличие от *iptables*, здесь нет предварительно определенных таблиц или цепочек, таких как фильтр или INPUT. Вместо этого *nftables* распознает базовые цепочки, прикрепленные к ядру, и не-базовые цепочки, которые являются просто контейнерами для правил. Не-базовые цепочки доступны только из базовых (через переходы *jumps*) и нужны разве что для поддержания порядка в вашем наборе правил.

Однако это не единственное отличие. Цепочки в *nftables* могут иметь несколько привязанных к ним действий, но по умолчанию у них может не быть счетчиков пакетов/байт. Это делается по соображениям производительности. Приращение целого числа каждый раз, когда пакет выполняет правило, не выглядит большой бедой; однако большинство компьютеров в наши дни многопроцессорные, поэтому надо заботиться о синхронизации этих приращений. В некоторых сценариях эти т.н. издержки синхронизации становятся заметными, так что стоит постараться избежать необходимости платить за подобные издержки, если только в этом нет особой необходимости.

Вот как создается таблица с помощью *nftables*:

```
$ sudo nft add table inet filter
$ sudo nft list tables
table inet filter
```

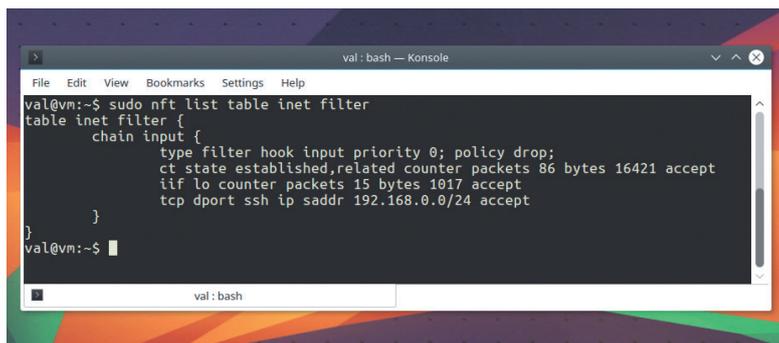
Мы называем это *filter* и ставим таблицу в семейство *inet*-адресов, сочетающее IPv4 (семейство *ip*) и IPv6 (*ip6*).

Раньше за IPv4 и IPv6 отвечали отдельные инструменты, что было неудобно на хостах с двойным стеком.

Вот как в эту таблицу добавляется базовая цепочка:

Wiki *nftables* — авторитетный ресурс для всех ваших нужд по части *nftables*.





```
val@vm:~$ sudo nft list table inet filter
table inet filter {
  chain input {
    type filter hook input priority 0; policy drop;
    ct state established,related counter packets 86 bytes 16421 accept
    iif lo counter packets 15 bytes 1017 accept
    tcp dport ssh ip saddr 192.168.0.0/24 accept
  }
}
val@vm:~$
```

Синтаксис, используемый для написания скриптов nftables, такой же, как в списке твблиц nft, естественно, кроме значения счетчика.

```
$ sudo nft add chain inet filter input { type filter hook input priority 0 \; policy drop \; }
```

Опять же, имя может быть любым. Цепочки в nftables принадлежат к разным типам, например, filter, route или nat. Тип определяет, как пакеты перемещаются по цепочке. Например, предположим, что только первый пакет в потоке идет в таблицу nat, при этом цепочки filter видят каждый пакет. Тип таблицы поддерживает различия между семьями: например, в `arp`: нет route.

Ловушки и приоритеты напрямую проецируются на свои аналоги в ядре. Здесь мы прикрепляем цепочку к той самой ловушке, которую мы видели в вышеупомянутой функции `ip_local_deliver()`. Wiki nftables (<https://wiki.nftables.org>) дает список возможных ловушек и типичных значений приоритетов, но вы всегда можете посмотреть `enum nf_ip_hook_priorities` и т. п. в исходниках ядра, чтобы составить самую обновленную картину.

Причина, по которой вы не можете видеть здесь произвольные последовательные значения (1, 2, 3), заключается в том, что этот порядок приоритетов соединен с другими подсистемами ядра, такими как conntrack или NAT. На самом деле они не являются частью iptables (как оказывается, это общее заблуждение), поэтому nftables использует их без проблем.

Политика — это вердикт для пакетов, не удовлетворяющих правилам. Она опциональна и по умолчанию установлена

»» BPFILTER: НОВИЧОК В КЛАССЕ

Nftables часто рассматриваются как iptables нового поколения, что по большей части правда. Однако в мире ИТ всё очень быстро меняется, и может устареть, не успев выйти на широкое использование.

Мы не знаем, произойдет ли это с nftables. Но во время написания этой статьи в сообществе Netfilter обсуждалось нечто под названием bpfiler. В феврале 2018 г. проект находится на ранней экспериментальной стадии. Однако в списке автора уже находятся Алексей Старовойтов [Alexei Starovoitov], главный в eBPF, и Дэвид С. Миллер [David S. Miller], специалист по сетевым подсистемам Linux; и это подтверждает, что bpfiler — не пустая забава.

И nftables, и eBPF реализуют основанную на ядре виртуальную машину. В последнее время в eBPF было вложено немало труда (см. **LXF232**), что дало ему такие функции, как компиляция Just-in-Time (JIT, «точно в срок»), проверка кода и даже разгрузка оборудования. Использование всего этого для фильтрации пакетов выглядит вполне естественным шагом. Текущее доказательство концепции привязано к XDP («как можно раньше»), но остальное может быть добавлено в будущем.

Никто не оспаривает идею использования VM eBPF. Однако текущий bpfiler эмулирует старый API iptables. Это беспокоит Харальда Вельте [Harald Welte], основного разработчика Netfilter/iptables, который считает это шагом назад и советует создавать bpfiler на API nftables. Дэвид Миллер возражает, говоря, что iptables пока что более популярны, так что это увеличивает объем тестирования.

на «accept». Заметьте, что позднее цепочки с низким приоритетом могут удалять принятые пакеты, но не наоборот (несмотря на то, что может говориться в документации). Более того, обратите внимание, что код вверху — это команда оболочки — вот почему надо заковытить точку с запятой.

Добавление не-базовой цепочки проще — вы просто не присоединяете ее к ловушке:

```
$ sudo nft add chain inet filter ssh
```

Однако самое потрясающее и заметное изменение — это, конечно, синтаксис. Если iptables используют топорный стиль `getopt()` — `-p tcp --dport 80`, nftables предлагают куда более читаемый DSL, позаимствованный из фильтров `tcpdump`.

Красивый синтаксис

В мире iptables вполне обычно иметь большой скрипт для настройки брандмауэра правило за правилом. Многие инструменты движка, такие как Shorewall (<http://shorewall.org>), идут по этому пути. Проблема в том, что установка правил таким образом не является единой операцией, и на долю секунды ваш набор правил оказывается в наполовину настроенном состоянии. Может случиться, например, что порты, которые должны быть закрыты, окажутся открытыми, что в итоге может привести к бреши в системе безопасности.

Совсем иначе это происходит в nftables. Во-первых, изменения в том, как правила хранятся внутри ядра, упрощают атомарную замену правил. Во-вторых, вместо того, чтобы давать множество команд, вы храните набор правил в файле, который вы передаете в `nft -f`. Эта команда загружает новые правила в ядро, а потом делает обмен [swap], так что вы никогда не увидите брандмауэр настроенным наполовину. Набор правил очень похож на скрипт, и на самом деле вполне нормально сделать его исполняемым, прислав символы «шалаша» — `#!`:

```
#!/usr/sbin/nft -f
# Ваша настройка может покрывать много правил
include "common.nft"
define local_net = 192.168.0.0/24
define bittorrent_ports = { 6881-6889, 6969 }
table inet filter {
  chain input {
    type filter hook input priority 0; policy drop
    ...
  }
}
```

Вы видите, что поддерживаются включения, переменные и комментарии. Чтобы обозначить переменную, вы предваряете ее имя символом `$`, как в `Bash` или Perl.

Обратите внимание на фигурные скобки в определении `"bittorrent_ports"`. Так вы создаете набор, который является основной структурой данных в nftables. Наборы — это способ работы nftables с множественными значениями, такими как IP-адреса. Они рассматриваются как единое целое — мы увидим пример далее. Они хорошо работают для разрозненных значений, но у nftables есть и интервалы для непрерывных диапазонов: посмотрите внимательно на первый элемент в `bittorrent_ports`.

Для iptables ipset предлагает такие же возможности; здесь он идет встроенным, как гражданин первой гильдии. Есть также карты [maps], словари [dictionaries] (также известные как карты суждений [verdict maps]) и сцепления [concatenations], но они являются расширенными функциями, о которых сегодня мы говорить не будем.

Выше показан первый синтаксис, который поддерживает nftables, и именно это вы получаете, когда загружаете в таблицу командой `nft list table`. Еще один пример (ниже) — это синтаксис, который мы используем для примеров командной строки, без префикса `nft`:

```
#!/usr/sbin/nft -f
add table inet filter
add chain inet filter input { ... }
```

Это не то же самое, что несколько команд `nft` (помните об атомарности). Более того, эти два синтаксиса не исключают друг друга. Довольно часто вы начинаете с `add table` с последующим `flush table`, чтобы начать заново. Иначе правила в вашем скрипте сольются с теми, что в данный момент находятся в ядре, а это вряд ли будет желательным результатом.

Биты важны, потому что они создают инфраструктуру. Однако инфраструктура не может отличить хороший трафик от плохого. Для этого надо добавить правила, т.е. выражения и соответствующие вердикты.

Выразите себя

Как и остальной синтаксис `nftables`, выражения читаются естественно, например, `tcp dport 22` или `ip saddr!=127.0.0.1`. Каждое выражение начинается с типа, за которым следует значение для проверки. Тип может быть протоколом, например, `ip`, `ct` для операций `conntrack` (брандмауэр с отслеживанием состояния) или `meta`. Последний облегчает проверку мета-информации: сетевых интерфейсов, отметок пакетов и т.д. Для значений поддерживаются простые арифметические операции и операции сравнения, а также некоторые сложные операции, например, хэширование. У вас может быть более одного выражения на каждый вердикт; в этом случае для соответствия правилу все они должны быть истинными.

Для вашего удобства `nftables` определяют символические имена для стандартных констант, например, флагов TCP. Сравните `tcp flags & (syn | ack) == syn` с `tcp flags & 0x6 == 0x2`; что лучше? Эти константы связаны с типами данных, и вы можете получить всю информацию из самой `nftables` командой `nft describe`:

```
$ sudo nft describe ct state
ct expression, datatype ct_state (conntrack state)
(basetype bitmask, integer), 32 bits
pre-defined symbolic constants:
invalid 0x00000001
new 0x00000008
...
```

Попробуйте это с другими выражениями, когда будете их встречать, чтобы понять внутреннюю работу `nftables`. Более подробную информацию вы найдете в man-странице `nft(8)`.

Вот так может выглядеть базовый набор правил рабочей станции:

```
ct related,established accept
tcp dport ssh ip saddr $local_net accept
```

» УХОД ОТ IPTABLES

`Nftables`, возможно, выглядят мило, но у вас, как у администратора, могло накопиться немало скриптов настройки, использующих `iptables`. И как хороший администратор, вы вряд ли перейдете на что-то новое, если это означает необходимость переписывать всё с нуля.

Это важные проблемы, и `nftables` предлагает возможность перехода для тех, кто работает на брандмауэре `xtables`. Последнее — это лишь хитроумный способ сказать “`iptables`, `ip6tables` и `ebtables`” одним словом, так что если вы используете какие-то из них — поздравляем, о вас позаботились.

`Nftables` рассматривает переход под двумя углами. Один — перевод правил, и вы можете применить его ко всему набору правил, созданному с помощью `iptables-save` или одиночным правилам. Последнее — хорошая возможность изучить `nftables` на примере: вы вводите синтаксис `iptables` и получаете эквивалент в `nftables`:

```
$ sudo iptables-translate -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
nft add rule ip filter INPUT ct state related,established counter accept
```

Эта функция должна быть доступна для дистрибутивов с `iptables` 1.6.1 или выше; если у вас `Ubuntu Xenial`, вам не повезло.

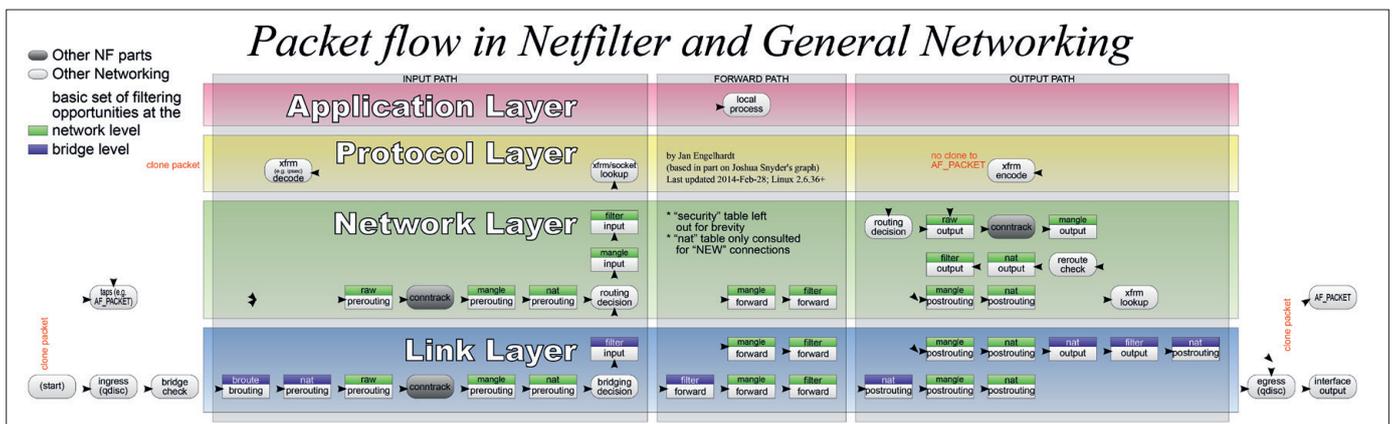
Еще одна опция — использовать движок `compat`. Вы вызываете команду `iptables-compat` вместо простого `iptables`, и правила переводятся прозрачно. Обратите внимание, что некоторые функции `iptables` могут отсутствовать, и смешивать и совмещать `iptables-compat` и `iptables` — не лучшая идея.

Последняя ремарка: эти инструменты — только помощники. Очень мелкие различия в операции могут быть критически важными для безопасности, так что всегда найдите время, чтобы оценить результат.

Вставьте это в приведенный выше скрипт вместо многоточия, чтобы завершить пример. Здесь мы разрешаем установленный и связанный трафик, чтобы рабочая станция могла определить происхождение исходящих соединений по своему усмотрению. Это не является безопасным, но вполне нормально для большинства случаев. Затем мы включаем SSH-соединения из локальной сети. В реальной жизни вам также понадобится разрешить трафик `localhost`.

Теперь вы понимаете, как Linux фильтрует пакеты, из глубин ядра до синтаксиса с высокоуровневыми правилами. `Nftables` предлагает ряд преимуществ по сравнению со своими предшественниками; атомарность, гибкость и читаемость — лишь некоторые из них. Но помните, что это вы, а не инструменты и их синтаксис, делаете систему безопасной. Всегда следуйте оптимальным процедурам безопасности и будьте защищены! **LXF**

Эта схема движения пакетов Netfilter. Она использует название `xtables`, однако общая схема действительно для всего остального.



» 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

Учебники



Евгений Балдин
Подтвердивший
свою квалификацию
физик.

» GITHUB, ДАВАЙ ДО СВИДАНИЯ!

Мы, злодеи, так просто не помираем.
*Рэви, Эпизод 12, 2-й сезон,
«Пираты Черной лагуны»*

Одна известная во всем мире компания сходила в магазин и на сдачу в семь с половиной миллиардов долларов прикупила себе Github. Вполне ожидаемо тема вошла в турбулентность. Самопальные опросы демонстрируют, что более половины держателей репозиторий свободного ПО на Github готовы уйти куда глаза глядят. Такая вот репутация покупателя в данной среде.

Github — это не просто сервер с Git. Это своеобразная социальная сеть, которая живет благодаря весьма разнородной толпе. Возможно, конкретно эта толпа денег особо и не приносит, ноет и капризничает кто во что горазд, но без нее в этом месте была бы электронная пустыня.

Тут же возник подряд на утешателей сообщества во главе с исполнительным директором The Linux Foundation Джимом Землином [Jim Zemlin]. Делаются разумные утверждения, что покупатель нынче уже не тот и не считает GPL, как ранее, раком программного рынка. Покупатель теперь хочет всяких разработчиков, включая разработчиков свободного ПО, поэтому мамой клянется, что покупку портить не будет, а будет ее только развивать.

Я не думаю, что сразу все будет уничтожено, да и будет ли вообще. Но с Github в обозримом будущем уйду. Не сказать, что от этого Github обеднеет, ибо я не программист, а пользователь, но свой интерес стараюсь блюсти. В этом не очень понятном случае единственное, что мне остается — следовать по направлению, указываемому этикой Свободного ПО, потому что будущее пока от нас сокрыто.
E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...

MOREUTILS

Улучшать продуктивность 56
Шашанк Шарма откопал коллекцию простых инструментов, которые не потребуют много времени на изучение и сделают работу с оболочкой приятным занятием.

УПРАВЛЕНИЕ ПРОЕКТОМ

Управлять проектом 58
При содействии **Бобби Мосса** воплощайте свои идеи в жизнь, предварительно облачая их в форму, способствующую практической реализации.

WORDPRESS

Создавать сервер 60
Сервер web-разработчика понадобился **Уиллу Мейстеру**, чтобы не выкатывать свои упражнения в тестировании на публику. Дуракам половину работы не кажут...

SYNCTHING+ENCFS

Настраивать облако 64
Стоит ли доверять посторонним свои конфиденциальные данные и терять над ними контроль? **Брайан Морк** уверен, что нет — уж лучше самому создать облачный сервис.

DOCKER

Развертывать приложения 68
Виртуальные машины уходят в прошлое: **Маянк Шарма** принимается за виртуальные приложения! И запускает их в контейнерах на почти «родной» скорости.

BUILDROOT

Автоматизировать сборку 72
Для создания встроенных систем **Дмитрий Кузнецов** выбрал подходящий инструментарий, с которым интегрировать зависимости — пара пустяков.

ALPINE+DOCKER

Докеризоваться 76
Евгений Гольшев делится накопленным опытом изготовления Docker-образов для ARM (и не только) на настольных компьютерах и ноутбуках.

АКАДЕМИЯ КОДИНГА

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

HUGO

Нуго, братцы, жить 80
Михалис Цукалос создает на Hugo сайты, состоящие из статических файлов, обойдясь без базы данных или других типов ресурсов. Оно, кстати, и надежнее.

WORDPRESS

Клонируем сайты 84
Исследуя возможности WordPress, **Кент Ельчук** наткнулся на мультисайты. Родительский сайт служит связующим звеном для разных мест либо категорий.

MOREUTILS

Сокровища командной строки

Малоизвестная коллекция полезных Unix-инструментов для увеличения продуктивности в терминале... завлекательно, говорит **Шашанк Шарма**.



НАШ ЭКСПЕРТ

Шашанк Шарма — выступающий в суде адвокат из Дели и ярый фанат Arch Linux. Он всегда в поиске странных экспонатов.

В свое время одной из причин растущей популярности Unix стала его прочная философия проектирования, которая включала набор правил, формирующих способ разработки Unix и его бесчисленных инструментов. Среди этих принципов — тот, что продолжает вдохновлять начинающих разработчиков: заниматься одним направлением и достигать достойных результатов.

Как немало говорилось, идея заключалась в создании инструментов и утилит, которые обеспечивали бы одну полезную функциональность и делали это хорошо. Многие из популярных инструментов командной строки, применяемые пользователями Linux, являются потомками инструментов-оригиналов, написанных для Unix на основе этого принципа проектирования.

Ориентируясь на простоту и эффективность таких утилит, Джой Хесс [Joey Hess] (<https://joeyh.name>) решает создавать и коллекционировать разные виды этих простых инструментов. Вы можете найти полный список на сайте проекта и даже назначить свои любимые инструменты для включения в пакет *moreutils*.

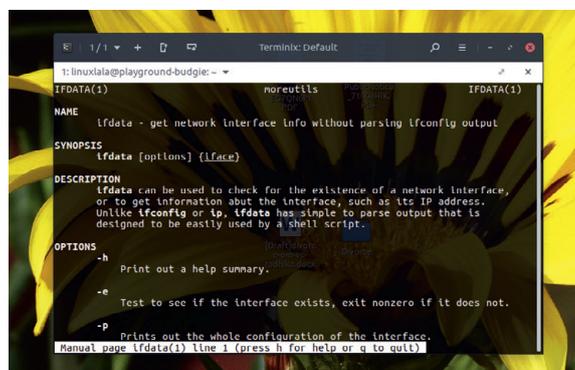
Пакет можно найти в репозитории ПО большинства дистрибутивов Linux. Запустите команду `sudo apt install moreutils` для установки его на Debian или Ubuntu и производных дистрибутивов. Для установки на Fedora понадобится команда `su -c 'dnf install moreutils`. Самая последняя версия *moreutils* — 0.62; однако большинство дистрибутивов в своих репозиториях ПО по-прежнему содержат версию 0.60 или старше. Если вам нужна последняя версия, заберите источник с репозитория проекта на Git и скомпилируйте его сами.

Откройте терминал и запустите команду `git clone git://git.joeyh.name/moreutils`. Затем, используя репозитории ПО, установите пакеты *docbook-xsl*. После этого перейдите в каталог, созданный командой Git, и запустите команды `make` и `make install`.

С момента написания в *moreutils* вошли порядка 15 различных инструментов по 4 категориям:

Категория	Инструменты
Работа с	pipesmispipes, pee, vipe, sponge
Обработка ошибок	errno, ifdata, isutf8, mispipe
Редактирование текстовых файлов	combine, isutf8, ts, vidir
Запуск программ	chronic, ifne, lckdo, parallel, zrun

К счастью, простота этих программ такова, что требуется лишь немного времени провести с любым из них, чтобы понять назначение и функциональность. Поскольку нам здесь не хватает места для подробного описания каждого из этих



Каждый из инструментов, включенных в *moreutils*, имеет свою man-страницу, но не все настолько детальны, как страница для *ifdata*.

Инструментов, мы рассмотрим несколько вариантов, начиная с инструментов редактирования.

vidir

Этим простым инструментом можно редактировать имена каталогов и файлов с удобством текстового редактора. Если каталог при вызове команды не указан, инструмент отображает содержимое текущего каталога в настроенном вами редакторе. Указанные файлы или содержимое каталога перечислены в виде нового файла *tmp*.

Поскольку все файлы *vidir* открывают файлы и каталоги в текстовом редакторе, не забывайте нажимать Сохранить, а затем выйдите из редактора, чтобы изменения вступили в силу. Так, если вы хотите в текстовом редакторе удалить файл из списка, соответствующий файл будет автоматически удален из вашей системы при сохранении и выходе из редактора.

Поскольку инструмент просто представляет имена файлов в текстовом редакторе, многие функции *vidir* зависят от назначенной в вашей системе переменной среды \$EDITOR. Например, с *vi* можно сделать больше, чем с *nano*.

combine

Можно использовать утилиту объединения для определения общих элементов в двух файлах с использованием нескольких логических операторов, таких как NOT, AND, OR и XOR.

Команда принимает два имени файла, разделенные логическим оператором, который вы хотите использовать для сравнения: `combine file1 <оператор> file2`. Если вам надо

СКОРАЯ ПОМОЩЬ

Узнать текущий текстовый редактор по умолчанию можно, запустив команду `echo $EDITOR`. Команда `export EDITOR="/usr/bin/vi` установит редактор по умолчанию как *vi*.

просмотреть только содержимое, общее для файлов **file1** и **file2**, вы можете сделать это с помощью команды `combine file1 and file2`. Используя логический оператор AND, инструмент будет выводить только общий текст в обоих предоставленных файлах.

Аналогичным образом можно использовать оператор NOT для получения содержимого, которое находится в **file1**, но не в **file2**. Оператор XOR может использоваться для содержимого, уникального для каждого файла.

Одним из интересных способов использования `combine` является вывод такой многословной команды, как `dmesg`. При первом запуске команды выведите вывод в файл: `dmesg > dmesg-output.txt`. При следующем запуске этой команды можно отослать ее вывод в `combine` и получить список изменений с момента предыдущего запуска `dmesg`:

```
$ dmesg > dmesg-output.txt
$dmesg | combine - not dmesg-output.txt
```

Если вы, например, между запусками этих двух команд подключились к USB-накопителю, вторая команда покажет только новое USB-устройство и ничего больше. Использование `-` во второй команде говорит о том, что надо читать **stdin** как первый файл, а `dmesg-output.txt` — как второй.

vipe

Если вы хоть сколько-нибудь поработали с оболочкой, вы вряд ли обошлись без канала `[pipe]` (`|`) для переключения стандартного вывода команды на ввод другой команды по умолчанию. Например, команда `ps aux | grep firefox` выделит запись, касающуюся *Firefox*, из всего вывода, созданного командой `ps aux`.

С помощью `vipe` можно отредактировать вывод первой команды, прежде чем она даст толчок второй команде. Синтаксис команды прост: `команда1 | vipe | команда2`. Продолжая пример, который мы только что привели, `ps aux | vipe | awk '{print $1, $2, $11}'` напечатает только три колонки, как указано команде `awk`, но сначала вывод откроется в настроенном вами редакторе, где вы сможете отредактировать вывод, а затем сохранить файл и выйти из редактора; текущий вывод будет передан на `awk`.

Как и другие инструменты, описанные выше, `vipe` создает только временные файлы, и вам незачем волноваться об их подчистке.

sponge

Другим полезным инструментом для работы с каналами является `sponge` [англ. губка], который «впитывает» стандартный ввод и записывает его в указанный файл. Это лучше пояснить на примере. Пусть у вас есть файл, содержащий пять несортированных чисел от 1 до 10. Шаги, связанные с сортировкой этого файла и сохранением содержимого в том же файле, будут следующими:

```
$ sort numbers.txt
1
2
5
7
9
$ sort numbers.txt > sorted-numbers.txt
$ mv sorted-numbers.txt numbers.txt
```

Здесь первая команда просто выводит отсортированные числа. Если вы хотите сохранить отсортированные числа в одном файле, следует добавить две последние команды. Однако с помощью `sponge` вышеупомянутое может быть достигнуто всего одной командой: `sort file-numbers.txt | sponge filenumbers.txt`.

Вместо создания временных файлов утилита `sponge` принимает весь стандартный ввод. Вы даже можете употребить вместе

»» MOPE MOREUTILS

Вот краткое введение в некоторые пакеты `moreutils`, которые делают работу с оболочкой приятной:

ts Используйте этот инструмент для добавления временной метки в начало каждой строки вывода `stdout`. Это поможет вам контролировать, сколько времени требуется на отработку команды, прежде чем передать управление следующей команде. Можно даже использовать инструмент `ts` для отметки времени журналов, созданных сервером.

zrun Текстовые редакторы не способны воспринимать сжатые файлы, но `zrun` может на лету распаковывать предоставляемые ему архивы, и в настоящее время работает с различными форматами, включая GZIP, BZIP2 и XZ. По умолчанию `zrun` удаляет распакованный временный файл после завершения, т.е. все изменения, внесенные вами в редактор, будут потеряны, если вы не сохраните содержимое вручную в новом файле.

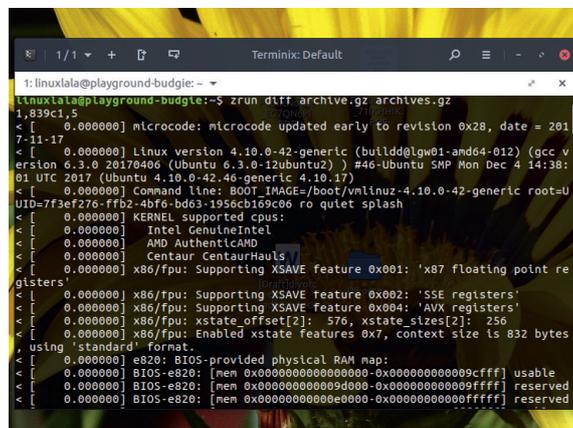
ifdata Существует масса инструментов, пригодных для получения информации о сетевых интерфейсах, количестве пакетов ввода/вывода и т.д. Один из самых популярных из них — `ifconfig`, но он часто создает данные, неподходящие для большинства пользователей, особенно если вы хотите вставить их в скрипт.

С помощью `ifdata` вы можете легко определить всю эту информацию и многое другое. Обратитесь к странице руководства для множества различных параметров команды, которые можно вызвать с помощью `ifdata`. Одной из основных является команда `ifdata -p wlp4s0`, она отображает конфигурацию указанного интерфейса.

со `sponge` параметр `-a`, если хотите добавить стандартный вывод в файл, вместо перезаписи файла: `sort file-numbers.txt | sponge -a filenumbers.txt`.

Мы рассмотрели лишь небольшую часть из 15 инструментов, которые в настоящее время составляют пакет `moreutils`. Ограничения на объем публикации препятствуют обсуждению этих инструментов еще более подробно, но, честно говоря, то, чего вы можете достичь с помощью этих инструментов, ограничено вашим собственным использованием и творчеством.

В отличие от многих других инструментов командной строки, имеющих массовую привлекательность, инструменты, включенные в `moreutils`, предназначены тем, кто активно работает со скриптами оболочки и т.п. Тем не менее, пользу и простоту этих инструментов невозможно переоценить. **LXF**



Команда `zrun diff archive.gz archives.gz` может сравнить два сжатых архива.

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

УПРАВЛЕНИЕ ПРОЕКТОМ

CherryTree:

ТОЛКОВЫЕ ЗАМЕТКИ

Бобби Мосс пришел помочь вам превратить цветы своих мыслей в сочные плоды, которыми вы накормите свои проекты.



**НАШ
ЭКСПЕРТ**

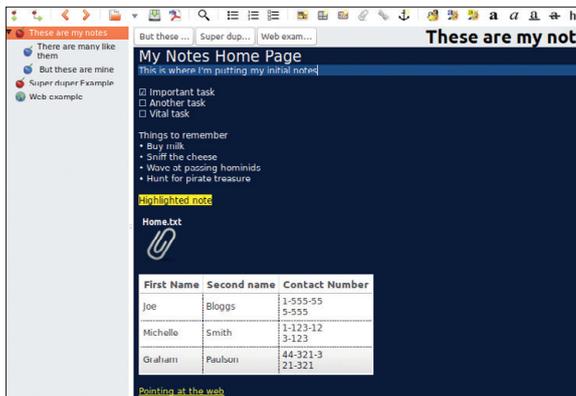
Бобби Мосс разрабатывает API промежуточного ПО для глобальной ИТ консалтинговой фирмы и ее клиентов. В свободное время работает над проектами свободного ПО, возится со старым оборудованием и записывает эксцентричные подкасты про Linux.

Все проекты начинаются с идеи. Но прежде чем воплощать идеи в жизнь, мы должны их сформулировать. Что мы знаем? Что нам надо исследовать? Какие шаги следует предпринять, чтобы наша идея реализовалась? Давайте сделаем драматическую паузу, чтобы обдумать эти вопросы...

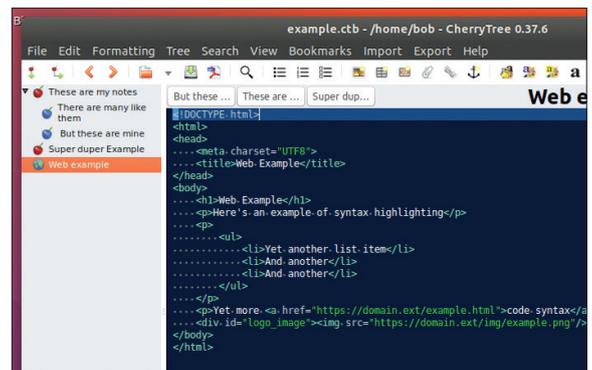
Не волнуйтесь, это не конференция TED, которую мы по ошибке перенесли на наши почтенные страницы. На нашем уроке мы рассмотрим приложения для заметок, и хотя это, возможно, не самые потрясающие инструменты в мире, вы найдете их полезными, например, при создании вашего шедевра для YouTube, для начала нового бизнеса, конспектирования лекции или написания вашего фундаментального труда.

Приложения для заметок не только компактно хранят полезную информацию: они помогают вам структурировать свои мысли в нечто реализуемое. Они также могут размещать ссылку на информацию на разных сайтах, на которую вы, возможно, захотите сослаться, и делают это более удобно, чем огромный и беспорядочный список старых закладок в вашем браузере.

Мы взглянем поближе на *CherryTree* [англ. вишневое дерево], потрясающее новшество в мире открытого кода, полное крутых функций для редактирования. Как вы, возможно, уже догадались по названию, оно также предлагает новаторский подход к организации ваших заметок в логичную древовидную структуру, чтобы вы могли разбить сложные идеи на более простые составные части.



Редактор форматированного текста *CherryTree* умеет работать с таблицами, встроенными файлами и списками ошибок, украсив это и без того прекрасное приложение для заметок.



Разработчики в *CherryTree* не обижены: можно создавать заметки с выделением синтаксиса для самых разных языков программирования.

Ветвление

Кстати о логичной древовидной структуре: первое, что вы заметите при создании вашего первого узла — маленькая вишенка рядом с ним. Каждый «узел» в данном случае — ветвь дерева с прикрепленным к нему фрагментом текста. Щелкнув правой кнопкой по этому узлу, вы можете добавить узлы, дублировать узел, который редактируете в данный момент, или создать подузлы.

По мере раскрытия вашей идеи вы можете обнаружить, что верхний узел предлагает всеохватывающую информацию, тогда как подузлы являются меньшими компонентами, написанными более подробно. Например, родительский узел для приготовления пирога может содержать рецепт с разными ингредиентами, количествами и временем выпечки.

Однако этим вы не ограничены в структурировании своих идей. Например, щелкнув правой кнопкой по Insert today's node [Вставить сегодняшний узел], вы можете вести ежедневный журнал состояния работ над каждым проектом с отметками времени, и коль скоро они создаются под одним и тем же узлом, они будут сгруппированы вместе. Вы также можете сделать цветовую отметку разных проектов в свойствах узла, чтобы проще находить их впоследствии.

Готовы для текста

Создав свои узлы, вы увидите, выделяя каждый из них, что у вас имеется текстовый файл с именем узла в качестве названия на правой нижней панели. По умолчанию это всегда

СКОРАЯ ПОМОЩЬ

Если *CherryTree* нет в менеджере пакетов вашего дистрибутива, вы можете скачать программу установки и tar-архив с исходным кодом оригинала с www.giuspen.com/cherrytree.

форматированный текст, но можно выбрать и Plain text [Простой текст], и Automatic syntax highlighting [Автоматическое выделение синтаксиса] — для проектов кода.

К сожалению, на момент написания основная панель инструментов выглядит всегда одинаково, независимо от выбора опции при создании узла. Т.е. по нажатию на опцию, работающую только для узлов форматированного текста, первое, что вы получите — сердитое сообщение о том, что она не поддерживается. А к счастью, *CherryTree* пока еще находится в активной разработке, поэтому интерфейс может стать немного более интуитивным, когда вы будете пробовать его сами.

Все обычные опции форматирования, полагающиеся текстовому редактору типа *LibreOffice Writer*, присутствуют в основной панели инструментов. Они позволяют создавать разные уровни заголовков, менять шрифты и делать в ваших заметках выделение цветом, выравнивание и форматирование.

Кроме того, вы увидите, что можно создавать три разные типа списков из панели инструментов с помощью трех значков справа от увеличительного стекла. Редактор достаточно умен, чтобы сообразить, когда вы пытаетесь создать маркированный или нумерованный список, и именно здесь вы будете создавать списки своих задач.

Следующая группа значков предназначена для вставки в ваши заметки внешних элементов. Если вы хотите сделать ссылку на схему или таблицу по текущей теме, с помощью соответствующих значков можно вставить изображение или файл. Можно также использовать инструмент гиперссылки вместе с привязками страниц для создания подробных страниц контента к более длинным заметкам, или даже быстро «прыгать» по соответствующим заметкам, если надо.

Еще один инструмент, достойный внимания — инструмент вставки таблиц [Table insert]. Щелчок по этому значку на панели инструментов открывает окно, позволяющее определить его размер или импортировать в вашу заметку данные из внешнего CSV. Это бывает удобно, когда вам надо отследить реальные факты, необходимые для проекта, или даже суммы денег, потраченные вами на вашу идею.

Группы заметок

Пока что мы рассмотрели, как создавать и редактировать отдельные заметки вручную; ну, а если вам надо переместить данные между несколькими приложениями для создания заметок?

Вы будете рады узнать, что *CherryTree* хорошо подготовлено и для этой цели. В меню импорта вы можете увидеть огромный список приложений для заметок, из которых вы можете импортировать старые данные.

Однако импортер — инструмент двойного назначения: его можно также употребить для группового импорта новых данных в *CherryTree* вместо необходимости вручную создавать узлы и вводить в них всё с нуля. Просто выберите опцию Import folder of plain text files [Импортировать папку простых текстовых файлов]. Она укажет вам на узел родительского дерева, который надо использовать в качестве отправной точки, и затем создаст новую древовидную структуру на основе того, как вы разместили свои текстовые файлы и подпапки. Затем вы сможете вручную отсечь ненужные ветви и отредактировать дерево по своему усмотрению. Аналогично можно импортировать данные HTML с сайтов, если вместо этого использовать опцию Import folder with HTML files [Импортировать папку с файлами HTML]; процесс точно такой же.

Чтобы извлечь данные из *CherryTree* для применения в других программах, имеется универсальное меню Export [Экспорт],

которое выдаст все ваши заметки в виде сайта, документа PDF или папки с простыми текстовыми файлами, по вашему выбору.

Однако самая полезная функция в *CherryTree* — его универсальные инструменты поиска, которые помогут вам быстро и легко определить местонахождение и отобразить нужные заметки вместо того, чтобы каждый раз вручную просматривать всю древовидную структуру.

В меню Search [Поиск] можно выбрать поиск по содержимому текущего узла, по всей структуре дерева или просто найти узел с определенным именем. Кроме того, нажав Ctrl + F на любом узле, вы можете осуществлять поиск по регулярным выражениям или по целому слову внутри текущего узла, или ограничиться только выделенным в данный момент текстом.

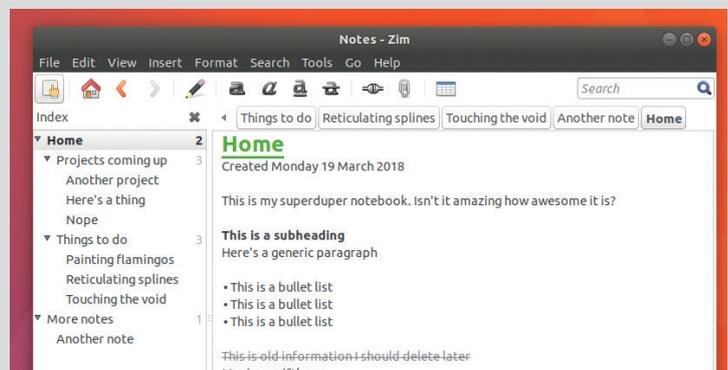
И, наконец, вы заметите отображение в виде навигационной цепочки [breadcrumb view] вверху панели текстового редактора. Оно поможет перемещаться по недавно редактировавшимся узлам независимо от их расположения в вашей древовидной структуре, так что вам не придется каждый раз их искать. **LXF**

» АЛЬТЕРНАТИВЫ CHERRY

Если вы считаете, что *CherryTree* не слишком подходит для вашей работы, есть множество отличных альтернатив. Один из примеров — настольная wiki *Zim*, с более понятным и интуитивным интерфейсом, но немного меньшим количеством функций. *Zim* также сохраняет ваши заметки в папке, а не в отдельном файле .ctb, и это значительно упрощает публикацию ваших заметок на web-сервере, чтобы другие могли их просматривать без привлечения дополнительных программ. Как и *CherryTree*, *Zim* легко установить прямо из вашего менеджера пакетов.

Еще один отличный пример — *Turtl*, который позиционируется как свободный, с полностью открытым кодом, безопасный и конфиденциальный конкурент куда более известного *Evernote*. В отличие от *CherryTree*, это приложение синхронизирует ваши заметки на разных устройствах, используя симметричный ключ шифрования, доступ к которому есть только у вас. *Turtl* имеет клиентские приложения для Windows, Mac, Linux и Android для размещенных на нем сервисов, и на момент выхода в печать активно разрабатывается приложение для iPhone. Вы можете скачать клиент *Turtl* с <https://turtlapp.com>.

SimpleNote — похожий проект; он начинался как приложение для iPhone, но превратился в сервис с рабочими клиентами на всех основных рабочих столах и мобильных платформах. К сожалению, пакета RPM нет, так что пользователям Fedora и openSUSE придется компилировать из исходника или устанавливать имеющийся пакет Debian с *alien*. Однако его всё же стоит скачать (с <https://simplenote.com>) и повозиться с ним, благодаря его минималистскому интерфейсу, отличным инструментам поиска и поддержке разметки.



Настольная wiki *Zim* предлагает урезанную альтернативу *CherryTree* с очень простой публикацией на дисках общего пользования и web-серверах.

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

Строим сервер разработки для тестирования

Уилл Мейстер применил фреймворк LEDE/OpenWrt, дешевый походный роутер и немного смекалки для создания удобного сервера разработки.



НАШ
ЭКСПЕРТ

Уилл Мейстер создал свой первый сайт, когда была популярна *Mosaic 0.9*. Он продолжает считать, что HTML — лучшая вещь со времен начала продаж хлеба в нарезке.

СКОРАЯ
ПОМОЩЬ

А вам точно нужен на вашем сервере разработки демон FTP? В контексте локальной сети *secure sftp* легче устанавливается и более удобна!

Успешная перезаливка скоро установит LEDE на ваш роутер.

Порывная статья Джона Лэйна в **LXF234** показала, как далеко продвинулся за последние десять лет встроенный Linux. Благодаря высокопроизводительным системам на чипах (SoC), мы окружены оборудованием, способным работать под нашей любимой ОС. Возникает вопрос: почему так мало людей программируют умные устройства?

Одна из причин может быть в том, что большинство из нас не осознано, насколько это возможно. Введение Джона показало, как LEDE/OpenWrt может включить новые, творческие приложения для домашнего роутера. Наш урок стартует с места, где он остановился, с пошаговыми инструкциями по переназначению дешевого походного роутера в сервер разработки. Кроме того, заодно мы применим некоторые основные концепции LEDE/OpenWrt.

Кому нужен сервер разработки?

Сервер разработки — важный инструмент профессионального web-разработчика. Такие инструменты позволяют проектировщикам моделировать сложные взаимодействия, скажем, с поддержкой социальных сетей, не рискуя опозориться публично. Полупрофессиональные разработчики, как правило, нервно воспринимают создание таких систем, как бы они ни старались выйти за рамки статических страниц. А если их сервер разработки был бы дешевле пары джинсов, и достаточно мал, чтобы уместиться в кармане?

Мы собираемся создать сервер разработки с поддержкой LEDE/OpenWrt на крошечном «походном роутере» стоимостью £20 или £30. Мы используем обычные ресурсы для обеспечения функциональности LAMP. (LAMP — это аббревиатура для

стандартной комбинации Linux с web-сервером *Apache*, базой данных *MySQL* и языком скриптов на стороне сервера PHP, хотя ресурсоемкий *Apache* нам понадобится заменить.) И, как грандиозный финал, мы установим и настроим программное обеспечение WordPress для ведения блога. Когда речь идет о получении наибольшей отдачи с вашего доллара, лучше этого нет ничего.

Выбор оружия

Во-первых, надо выбрать роутер. На странице совместимости LEDE/OpenWrt представлено около тысячи устройств, многие из которых давно не производятся. Мы выбрали GL.iNet MT300A, потому что он объединяет 128 МБ встроенной ОЗУ и почти пристойный процессор MediaTek с низкой ценой и малым размером. Он удовлетворяет минимальным требованиям для нашего применения.

Мы установим множество пакетов, и если вы не знакомы с *PuTTY* и *scp* (*secure sftp*), изыщите время на прочтение руководства (они дружелюбны). Отнюдь не весело бороться с незнакомым ПО на вашем собственном компьютере в попытках спасти удаленную машину!

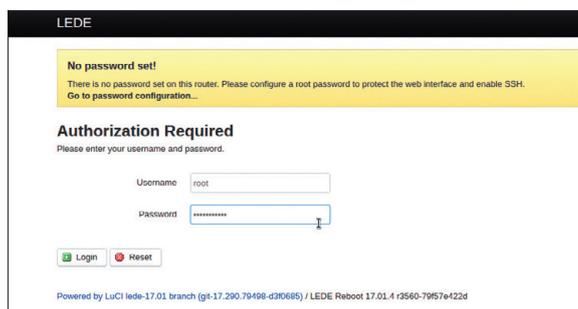
Чтобы убедиться в правильной работе нового роутера, следуйте инструкциям производителя для начальной настройки, но не поддавайтесь на удобство приятных экранов конфигурации. Нашим первым шагом будет замена программного обеспечения OEM на типовой LEDE/OpenWrt.

Проверьте свое устройство по таблице совместимости на сайте OpenWrt, скачайте соответствующую прошивку на свой компьютер, затем следуйте инструкциям web-интерфейса роутера, чтобы загрузить ее и перепрошить. (Для обновления нашего MT300A мы использовали **lede-17.01.4-ramips-mt7620-g1-mt300a-squashfs-sysupgrade.bin**.) Роутер перезагружается в IP-адрес LEDE по умолчанию **192.168.1.1**. Для доступа через графический интерфейс LEDE LuCI укажите браузеру новый адрес.

Теперь самое время подключить порт LAN нового роутера к имеющемуся роутеру коротким кабелем Ethernet и установить запоминающийся новый пароль **root**.

Получение доступа

Давайте настроим новый роутер на доступ к Интернету. Именно здесь потребуются навыки работы в командной строке! Запустите терминал на своем компьютере и выполните такие действия:



```
route -n
```

Отклик будет выглядеть примерно так:

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 192.168.1.254 0.0.0.0 UG 1024 0 0 wlan0
169.244.0.0 0.0.0.0 255.255.0.0 U 1000 0 0 wlan0
192.158.1.0 0.0.0.0 255.255.255.0 U 0 0 0 wlan0
```

Обратите особое внимание на настройку шлюза во второй строке. Это IP-адрес вашего имеющегося широкополосного роутера. (Широкополосные роутеры BT часто бывают установлены на 192.168.1.254). Вашему новому роутеру потребуется этот адрес для доступа к Интернету.

Запустите *PuTTY* с терминала и начните новую сессию.

Введите IP-адрес нового роутера (не шлюза) и войдите в систему как *root*, используя пароль, который вы задали ранее. Если вход выполнен успешно, вы увидите логотип *OpenWrt* в тексте и подсказку командной строки *[CL]*. Чтобы установить шлюз, введите команду *uci*:

```
uci set network.lan.gateway=192.168.1.254
```

В зависимости от настроек, вам может потребоваться набрать

```
uci set network.lan.dns=192.168.1.254
```

Затем сохраните изменения:

```
uci commit network
```

Перезагрузите новый роутер, затем снова войдите через *PuTTY* и используйте менеджер пакетов *OpenWrt opkg* для установки текстового редактора CLI *nano* таким образом:

```
opkg update
```

```
opkg install nano
```

Если вы получаете ошибки, то проблема — в конфигурации. Google вам в помощь! Загрузив *nano*, вы сможете просмотреть все параметры конфигурации сети за один проход, выполнив следующие действия:

```
nano /etc/config/network
```

LEDE/OpenWrt позволяет конфигурировать с помощью текстовых команд *uci* или редактированием файлов, как вам нравится.

Настройка web-сервера

Если вы столкнулись с процессом наложения (см. врезку внизу справа), вы оцените удобство *uHTTPd*, web-сервера LEDE/OpenWrt. Он запускает web-интерфейс LEDE и вполне может обслуживать ваши страницы. Мы поручаем *uHTTPd* прослушивать трафик на порте 81 и отвечать на такие запросы страницами из определенного каталога. В командной строке *PuTTY* введите:

```
uci set uhttpd.l1mp=uhttpd
```

```
uci set uhttpd.l1mp.listen_http=81
```

```
uci set uhttpd.l1mp.home=/srv/www
```

```
uci commit uhttpd
```

Затем, всё еще в *PuTTY*, введите

```
mkdir -p $(uci get uhttpd.l1mp.home)
```

Странного вида инструкция в скобках предписывает LEDE/OpenWrt создать новый каталог, соответствующий записи в файле конфигурации *uHTTPd*. Используйте команду *echo* для публикации web-страницы Hello World!:

```
echo "<html><body><p>OpenWrt — это круто!</p></body></html>" > /srv/www/index.html
```

Перезапустите *uhttpd*:

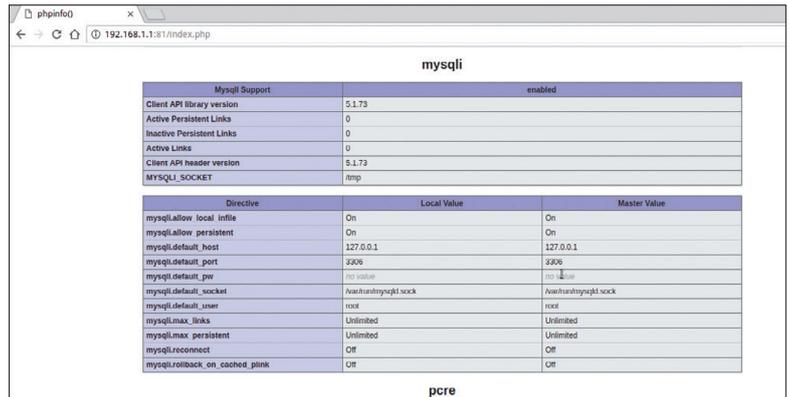
```
/etc/init.d/uhttpd restart
```

Теперь направьте браузер на 192.168.1.81. Вы должны увидеть сообщение, введенное вами несколько минут назад.

Установка MySQL и PHP

После установки и запуска web-сервера установка *MySQL* и *PHP* представляет собой последовательность команд *PuTTY*. Сначала *MySQL*:

```
opkg install mysql-server
```



Если страница состояния PHP показывает, что MySQL полностью настроен – значит, ваш стек LAMP запущен и работает!

```
sed -i 's,^datadir.*,datadir = "/srv/mysql",g' /etc/my.cnf
sed -i 's,^tmpdir.*,tmpdir = "/tmp",g' /etc/my.cnf
mkdir -p /srv/mysql
mysql_install_db --force
/etc/init.d/mysql start
/etc/init.d/mysqld enable
mysqldadmin -u root password 'ЗДЕСЬ_Д.Б._ВАШ_ПАРОЛЬ_ROOT_ДЛЯ_MYSQL'
```

Запустите демон *MySQL* и введите свой пароль:

```
root@Arduino:~# mysql -uroot -p
Enter password:
mysql> show databases;
```

Вы должны получить отклик такого вида:

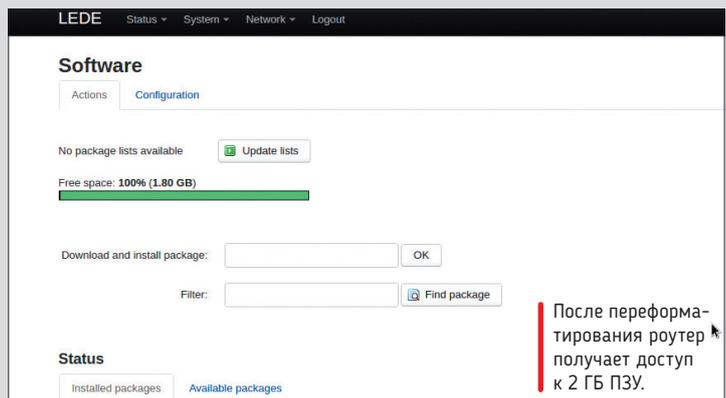
```
+-----+
| Database |
+-----+
| information_schema |
```

СКОРАЯ ПОМОЩЬ

При внесении изменений в OpenWrt быстрее перезагрузить соответствующий модуль, чем перезагрузить всю систему. Чтобы перезагрузить сеть, введите `/etc/init.d/network reload`.

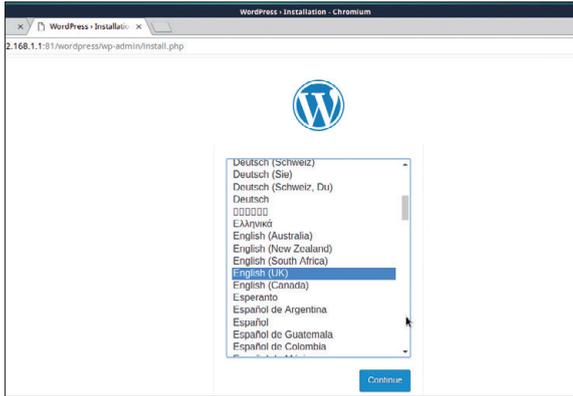
» У ВАШЕГО РОУТЕРА ЕСТЬ ПЗУ?

LEDE/OpenWrt крошечный, что и позволяет ему работать с мизерным хранилищем роутера. MT300A, описанный в этой статье, предлагает только 16 МБ флеш-памяти, и этого хватает для LEDE плюс стек LAMP, однако недостаточно для WordPress, который требует до 20 МБ. Если вы не хотите использовать более дорогое оборудование, вы можете расширить доступное ПЗУ вашего устройства с помощью метода, называемого наложением [overlying]. Это не для малодушных! Чтобы включить наложение, вам понадобится запасная USB-флешка, отформатированная в ext4. Учтите, что флешка не будет доступна для других устройств. Отформатировав флешку, следуйте инструкциям с <https://wiki.openwrt.org/doc/howto/extroot>.



После реформатирования роутер получает доступ к 2 ГБ ПЗУ.

Еще один блог WordPress намечается...



СКОРАЯ ПОМОЩЬ

Блоггеры полагаются на "permalinks" WordPress для создания запоминающихся URL-адресов, ориентированных на SEO. Однако в uHTTPd отсутствует функция `mod_rewrite`, встроенная в более тяжелые серверы, поэтому ваш сервер разработки не будет правильно обрабатывать эти постоянные ссылки. Но всё остальное с WP должно быть в порядке...

```

mysql |
test |
+-----+
3 rows in set (0.00 sec)

Просто введите exit в MySQL, когда закончите.
Вследствие экономного подхода LEDE/OpenWrt к использованию памяти, установка PHP сложнее:
opkg install php7 php7-cgi php7-mod-mysql php7-mod-pdo php7-mod-pdo-mysql

Затем используйте uci, чтобы добавить базовые установки PHP в файл конфигурации uHTTPd:
uci add _list uhttpd.lmp.interpreter='php=/usr/bin/php-cgi'
uci set uhttpd.lmp.index_page="index.html index.php"
uci commit uhttpd

Версия MySQL для OpenWrt не интегрируется с PHP автоматически, поэтому вам надо добавить целый новый раздел MySQL в файл настройки PHP — /etc/php.ini, соответствующий настройкам из файла /etc/my.cnf MySQL. Вот что было у нас:
[MySQLi]
mysqli.allow_local_infile = On
mysqli.allow_persistent = On
mysqli.cache_size = 2000
mysqli.max_persistent = -1
mysqli.max_links = -1
mysqli.default_port = 3306
mysqli.default_socket = /var/run/mysqld.sock
mysqli.default_host = 127.0.0.1
mysqli.default_user = root
mysqli.default_password = 'ЗДЕСЬ_Д.Б._ВАШ_ПАРОЛЬ_ROOT_ДЛЯ_MYSQL'
```

```

mysqli.connect_timeout = 60
mysqli.trace_mode = Off

Наконец, установите в /etc/php.ini пустое значение для переменной doc_root:
doc_root = ""
```

Это упростит доступ к PHP на вашем сервере разработки, хотя на производственной платформе мы настоятельно не рекомендуем так делать!

Чтобы протестировать новую установку, добавьте страницу `index.php` в каталог `/srv/www`:

```

echo "<html><body><?php phpinfo(); ?></body></html>" > /srv/www/index.php
```

Перезагрузите `uHTTPd` еще раз и укажите браузеру на `192.168.1.1:81/index.php`. Если вы видите длинную страницу статуса PHP со всеми ее записями MySQLi, ваш стек LAMP (или теперь уже LUMP?) готов.

Установка WordPress

Предполагая, что вы расширили флеш-память своего роутера (см. врезку на стр. 61), вы можете использовать свой новый стек LUMP в качестве платформы для WordPress.

Перейдите на <http://wordpress.org> и загрузите копию программного обеспечения на свой компьютер. Разархивируйте его и переименуйте `wp-config-sample.php` в `wp-config.php`.

Запустите терминал (не другой сеанс `PuTTY`!) и используйте `scp` для копирования распакованного WordPress в каталог `/srv/www` на новом роутере:

```

scp -r /home/[yermamaspc]/Downloads/wordpress/
root@192.168.1.1:/srv/www/
```

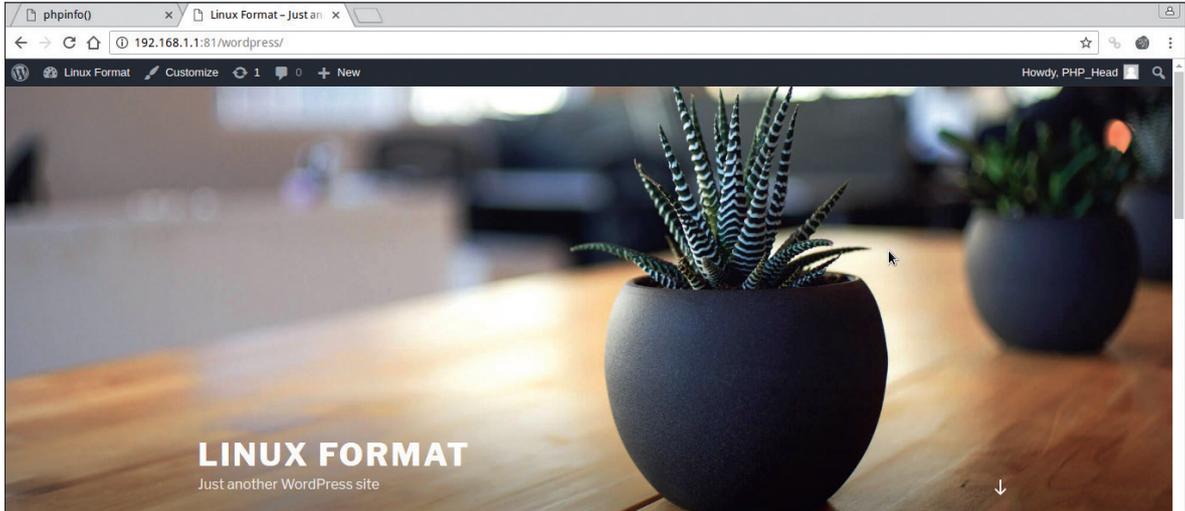
LEDE/OpenWrt не поддерживает PHP/MySQL, и вам надо вручную настроить базу данных MySQL для своего сайта WordPress. К счастью, это просто:

```

mysql -u root -p
CREATE DATABASE wp_dev;
CREATE USER 'wp_dev_user'@'localhost' IDENTIFIED BY
'ЗДЕСЬ_Д.Б._ВАШ_ПАРОЛЬ_ДЛЯ_WP_USER';
GRANT ALL ON wp_dev.* TO 'wp_dev_user'@'localhost';
```

Используйте `Putty` и `nano` для внесения соответствующих изменений в файл `wp-config.php` в `/srv/www/wordpress`, чтобы записи `DB_NAME`, `DB_USER` и `DB_PASSWORD` соответствовали тем, которые вы только что ввели в `mysql`.

Наконец, вызовите скрипт установщика WordPress из вашего браузера: `http://192.168.1.1:81/wordpress/wp-config.php`. Теперь вы можете рассматривать ваш новый сервер разработки как любую другую платформу WordPress, но не пропустите подсказку «Скорая помощь» про постоянные ссылки [permalinks]. **LXF**

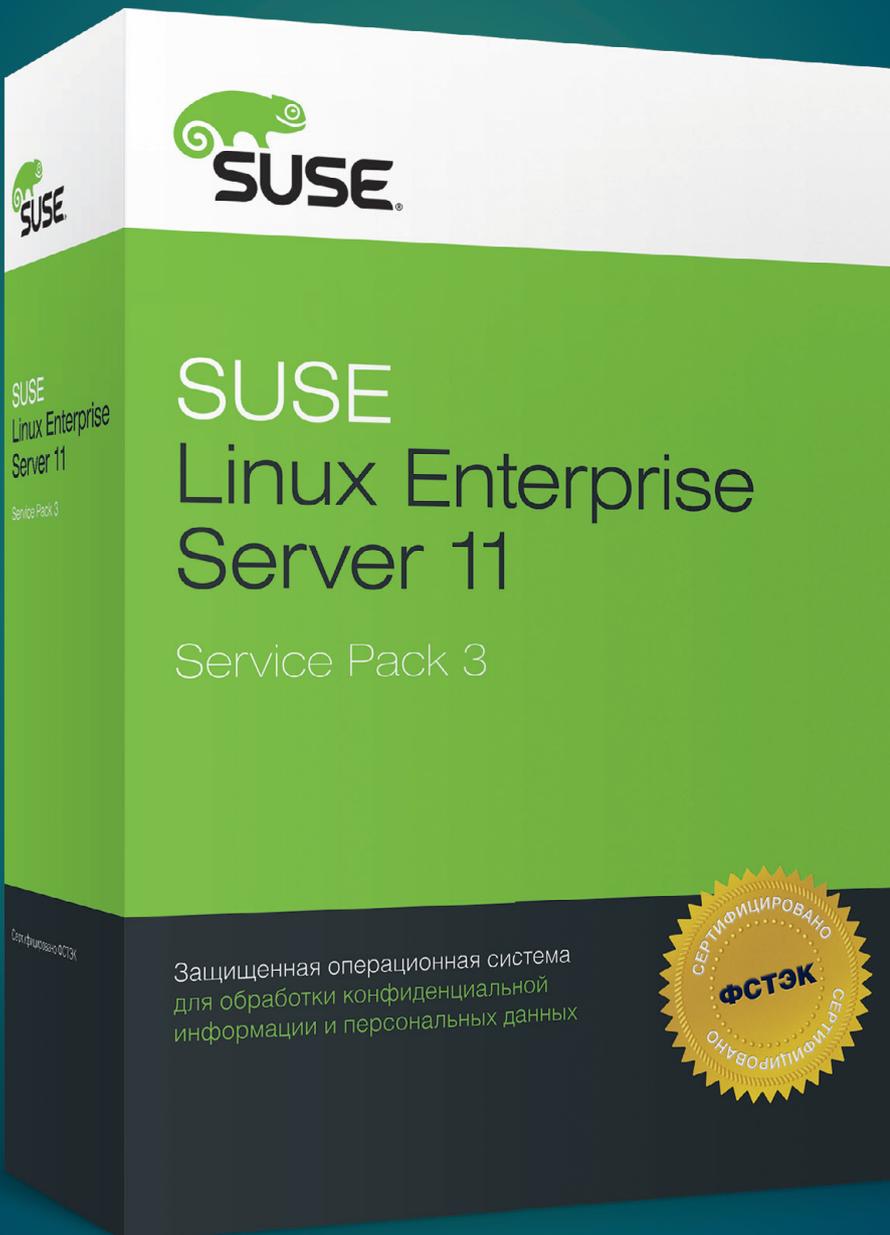


Вот первый пост WordPress на вашем сервере разработки, готовый завлечь вашу аудиторию.



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



SYNCTHING + ENCFs

Шифрованное облачное хранилище даром

Брайан Морк совершает прогулку в облаке — устанавливая и настраивая ПО, а также процедуры резервного копирования и опции безопасности и удобства.



НАШ ЭКСПЕРТ

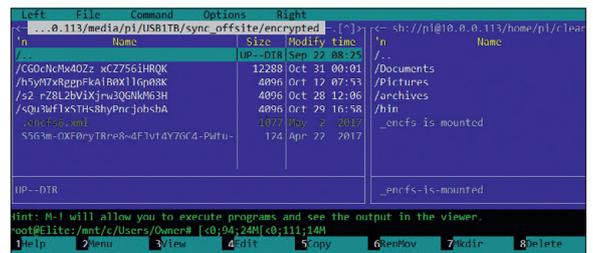
Брайан Морк пользовался Linux 23 года и написал десятки статей. Теперь он завален данными.

Облачное хранилище файлов является популярным вариантом. Коммерческие решения не хранят конфиденциальности, небезопасны, приносят в жертву ваш контроль над данными, часто требуют установки инвазивного программного обеспечения и включают покупку подписки. В части первой (см. **LXF236**) описана архитектура для создания бесплатного облачного сервиса резервного копирования, зашифрованного как при передаче, так и при хранении данных.

На нашем уроке рассматривается установка бесплатного облачного сервиса и управление им, в трех частях: настройка службы шифрования; настройка синхронизации; а затем работа с инструментами.

Установка EncFS

Шифруя файлы, EncFS сохраняет ту же структуру каталогов, отметки времени и размеры файлов. Это может позволить кому-то угадать, что это за файлы. Если кто-то получит несколько копий файла после небольших изменений или файл с известным содержанием, ему будет проще понять ключ шифрования. Плохой парень может даже заменить ваши данные старой версией, и вы не будете об этом знать, но это пресекут односторонняя



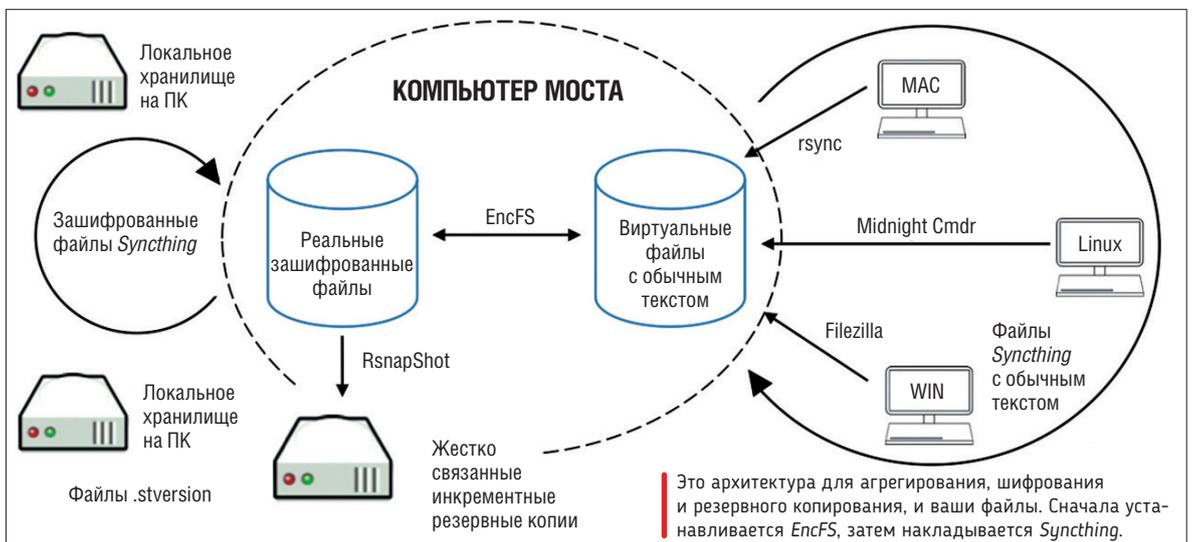
Midnight Commander показывает файлы, переданные на компьютер моста. Слева: эквивалентные зашифрованные файлы готовы для синхронизации вне сайта.

синхронизация с удаленными узлами и цель *syncthing* выявлять различия.

Однако попытка синхронизировать файлы, когда эти метаданные недоступны, затруднена. Например, рассмотрим том *Veracrypt*, который хорошо скрывает метаданные. Когда один файл изменен, невозможно войти в том и изменить один файл. Вместо этого весь том должен повторно синхронизироваться как один логический блок, а для получения файла понадобится

СКОРАЯ ПОМОЩЬ

Если вы не хотите вводить пароль при перезапуске EncFS и хотите сохранить свой пароль на компьютере, попробуйте опцию EncFS — `extrpass = 'cat encfspw.txt'`, где `encfspw.txt` хранит ваш пароль.



Это архитектура для агрегирования, шифрования и резервного копирования, и ваши файлы. Сначала устанавливается EncFS, затем накладывается Syncthing.

весь том. Так что ограничения *EncFS* смягчаются с помощью других методов.

На мостовом компьютере установите *EncFS*, используя стандартные команды *apt-get*. На момент написания статьи доступна версия 1.7.4. Первые две команды здесь не обязательны, но будут правильно сперва обновить настройки дистрибутива и программное обеспечение:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install encfs
```

EncFS в обычном (прямом) режиме делает виртуальные текстовые файлы доступными в указанной вами точке монтирования. Любые текстовые файлы, помещенные в точку монтирования, будут храниться в зашифрованном каталоге (и исчезнут из текстового каталога при остановке *EncFS*).

На компьютере моста создайте точку монтирования, где все ваши текстовые файлы будут видны и синхронизированы. Затем создайте каталог, где *EncFS* будет сохранять зашифрованные версии. Вы можете поместить зашифрованные файлы в подкаталог синхронизированного каталога вне сайта, но это не обязательно; они могут быть одинаковыми.

Для первоначальной настройки важно запустить *EncFS*, прежде чем *Syncthing* будет привязан к виртуальному текстовому каталогу. Дело в том, что когда *Syncthing* устанавливает ссылку, в корневом каталоге синхронизации создается файл семафора **.stfolder**. Вы хотите, чтобы этот файл был «захвачен» *EncFS* и распространен на зашифрованный каталог. Не позволяйте *Syncthing* создавать его в точке монтирования, когда *EncFS* не запущен, иначе этот семафор будет ложно присутствовать в поиске *Syncthing* — даже если *EncFS* не запущен. **.stfolder** никогда не должен находиться в базовой точке монтирования, иначе *Syncthing* может синхронизировать «ничто» со всеми узлами в кольце текстовой синхронизации (то есть удалить все файлы с ваших рабочих компьютеров!).

```
$ mkdir /home/pi/cleartextfiles
$ mkdir /mount/pi/USB1TB/sync_offsite/encrypted
$ encfs /mount/pi/USB1TB/sync-offsite/encrypted /home/pi/cleartextfiles
```

Creating new encrypted volume.

{... вывод программы обрезан ...}

Теперь надо будет ввести пароль для вашей файловой системы. Запомните этот пароль, так как абсолютно никакого механизма восстановления нет. Однако пароль можно изменить позже, используя *encfsctl*.

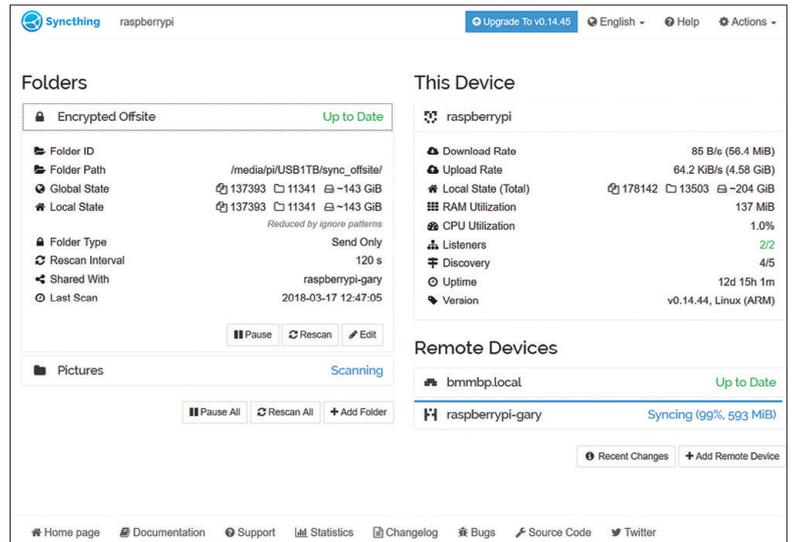
New Encfs Password:

Verify Encfs Password:

\$

При первом запуске *EncFS* создается файл конфигурации (текстовый **encfs6.xml**) на левой, зашифрованной стороне схемы (внизу стр. 64), а файл XML будет распространен за ваш сайт вместе с зашифрованными файлами. И это хорошо, поскольку этот файл и ваш пароль необходимы для дешифрования файлов! Возможно, вы захотите вручную сохранить этот файл в другом месте. Вы даже можете распечатать копию на бумаге. Он требуется для расшифровки файлов, но без вашего пароля он использоваться не может.

Автоматизация *EncFS* во время загрузки проблематична. Любой автоматический метод должен будет хранить ваш пароль где-то на компьютере моста. Вместо этого вручную запускайте *EncFS* при каждом запуске компьютера моста с помощью той же команды, что и при первом запуске. Когда файл XML будет найден, вам предложат ввести пароль. Если компьютер моста потеряет питание, то потребуются перезагрузка *EncFS*. Это можно сделать через SSH, если вы настроите SSH-сервер на запуск во время загрузки.



```
$ encfs /mount/pi/USB1TB/sync_offsite/encrypted /home/pi/cleartextfiles
```

EncFS Password:

Если вы хотите быстро увидеть, работает ли *EncFS*, просто перейдите в текстовый каталог и посмотрите, видны ли файлы. Если они есть, *EncFS* работает. Чтобы остановить *EncFS*, просто отмонтируйте просмотр текста:

```
$ sudo umount /home/pi/cleartextfiles
```

В программе командной строки *EncFSctl* есть дополнительные параметры, например, изменение пароля, связанного с файлом.

Установка Syncthing

Правая часть схемы (см. внизу стр. 64) выделяет несколько программ, которые можно использовать для объединения файлов с рабочими компьютерами на компьютер моста. *Syncthing* уникален среди опций, поскольку способен объединять файлы на компьютере моста за кулисами без ручного вмешательства. *Syncthing* распространяется как исполняемый файл без зависимостей библиотек. Полезно, что программа поставляется упакованной с инструкциями и примерами. Вы можете найти соответствующую загрузку на <https://syncthing.net>.

В файлах инструкций, поставляемых с дистрибутивом, и других онлайн-документах сайта довольно легко разобраться. Самая большая проблема может быть в том, где поместить исполняемый файл *Syncthing*. Можно просто-напросто положить его в свой домашний каталог. Если вы хотите получить больше помощи в установке *Syncthing*, для получения информации обратитесь к онлайн-документации.

Графический веб-интерфейс *Syncthing* на компьютере моста. 137 393 файлов в 11 341 каталогах, всего 143 Гб.

СКОРАЯ ПОМОЩЬ

Используя параметры командной строки, *Syncthing* может регистрировать многие действия. Если он включен, журнал отображается в меню Actions, Logs.

»» ВЫБОР ФАЙЛОВОЙ СИСТЕМЫ

Вы можете использовать емкий USB-накопитель для объединения ваших зашифрованных файлов перед внешней синхронизацией. Файловая система FAT32 будет наиболее совместимой (Linux, Mac, Windows). Это не сработало у нас, поскольку мы также хотели хранить в том же разделе репозиторий *Rsnapshot*, что требует жестких ссылок, которые FAT32 не предлагает. Кроме того, ограничение размера файла в 4 Гб в FAT32 может быть проблемой. В этом случае, возможно, вы предпочтете формат ext3. Если вы используете на съемном диске файловую систему Linux, помните, что надо позаботиться о правах доступа к файлам. Если USB-накопитель впоследствии будет употребляться в нескольких системах, у которых другие имена пользователей, вы можете обратиться к root для ручной смены владельца и информации о группе.

СКОРАЯ ПОМОЩЬ

Вы можете получить доступ к панели управления *Syncthing* с любого компьютера, подключенного к сети. Используйте вкладку *Actions*, *Settings* графического интерфейса, чтобы изменить 127.0.0.1 на любой IP-адрес, которому разрешено использовать графический интерфейс пользователя, включая 0.0.0.0:8384, доступный всем. Обязательно укажите имя пользователя и пароль!

Для компьютера моста на Raspberry Pi загрузите вариант для ARM. Есть несколько способов запускать его во время загрузки на основе вашей конкретной установки. Чтобы использовать автозапуск, вызовите менеджер рабочего стола *LXDE* и создайте файл запуска *Syncthing.desktop* в каталоге `~/config/autostart/`:

```
[Desktop Entry]
Type=Application
Name=Run Syncthing
Icon=
Exec=syncthing --no-browser
StartupNotify=false
```

Опять же, убедитесь, что *EncFS* запущен, прежде чем включить *Syncthing* для создания **.stfolder** во время его первого запуска и перед помещением чего-либо еще в текстовый каталог на компьютере моста. В точке виртуального монтирования текста ничего хранить не следует. Точка монтирования предназначена только для *EncFS*, чтобы захватывать файлы, шифровать их и хранить в другом месте.

Syncthing будет работать в фоновом режиме и собирать файлы с ваших рабочих компьютеров. В первой части этого учебника были описаны три различные парадигмы для выбора нужных файлов с вашего рабочего компьютера.

Вы поймете, что агрегация работает, когда в текстовом каталоге начнут появляться файлы. На самом деле они хранятся в зашифрованном виде в другом месте, но вы можете увидеть незашифрованную версию в каталоге агрегации.

Возьмите их на мост

Как только агрегация заработает, настройте дополнительное кольцо *Syncthing*, связывающее зашифрованный каталог компьютера моста с внешним компьютером (компьютерами). Для внешнего кольца синхронизации вы можете выбрать опцию *Send Only* в *Advanced Settings* на компьютере моста, чтобы никакие изменения на удаленном узле не возвращались к вам. Отправьте *Folder-ID* своему другу по электронной почте или

телефону на удаленный сайт и добавьте его удаленный компьютер в качестве авторизованного узла синхронизации. На удаленном сайте должен быть указан контроль версии "trash can [корзина]". Таким образом, если данные будут удалены из набора синхронизации, на удаленном сайте всё равно будет копия в каталоге **.stversions**.

Вы также можете использовать значок рабочего стола на вашем компьютере моста для запуска панели управления *Syncthing* через web-браузер. С помощью менеджера рабочего стола *LXDE* запишите следующее в файл *.desktop* управления *syncthing*:

```
[Desktop Entry]
Type=Application
Name=Syncthing Control Panel
Exec=epiphany 127.0.0.1:8384
Icon=
```

Чтобы остановить *Syncthing*, вы можете приостановить или удалить отдельные каталоги из графической панели управления. Используя графический интерфейс, также можно отключить локальный узел или остановить всю программу *Syncthing*. Чтобы остановить *Syncthing* без графического интерфейса, вы можете запустить монитор процессов, например, *top*, и просто убить процесс *Syncthing*.

Снимок экрана на стр. 65 показывает графический web-интерфейс *Syncthing* на компьютере моста. Каталог **photos** обновляет собранные данные с локальных рабочих компьютеров. За кулисами *EncFS* шифрует фотографии и хранит файлы в каталоге **Encrypted Offsite**, которым является смонтированный USB-накопитель на 1 ТБ. Каталог **Encrypted Offsite** актуален, потому что он не нуждается в файлах из других источников, а географически удаленный сайт резервного копирования под названием *raspberrypi-gary* на 96% синхронизирован с остатком в 3,19 ГБ.

Обратите внимание: каталоги **.stversion** и **.stignore** не передаются по ссылке *Syncthing*, потому что они являются метаданными, уникальными для каждого узла *syncthing*. Однако вы можете создавать копии с измененным именем файла, и *Syncthing* будет поддерживать их как «обычные» файлы.

» РУЧНАЯ АГРЕГАЦИЯ ФАЙЛОВ

При работающей *EncFS* вместо того, чтобы *Syncthing* выполнял агрегацию, вы можете агрегировать файлы в текстовом каталоге компьютера моста вручную. Просто убедитесь сначала, что *EncFS* работает, чтобы «поймать» их. Например, можно перетаскивать файлы на компьютер моста, используя *Bash*-скрипт *rsync* под Linux, а на Mac или Windows 10 — оболочку *Bash*:

```
set -x
ssh pi@10.0.0.113 test -f /home/pi/cleartextfiles/_encfs-is-mounted && rsync -rtv --delete-after ~/Documents/$1 pi@10.0.0.113:cleartextfiles/Documents
```

Команда *set* необязательна; она просто позволяет контролировать в окне терминала любые выполняющиеся команды *Bash*. Во второй строке используется функция *Bash* AND (&&). Команда *ssh* слева выполняет проверку, имеется ли файл семафора (**_encfs-is-mounted**) в нужной папке (аналогично **.stfolder** для *syncthing* на корневом уровне синхронизации). Это гарантирует, что *EncFS* запущена до копирования файлов. Если файл присутствует, правая сторона выполняет и передает обновления файла агрегатору с помощью *rsync*. Подкаталоги `~/Documents`, перечисленные в командной строке (или весь каталог `~/Documents`, если переменная *\$1* командной строки не введена) помещаются в каталог Pi **cleartextfiles**.

Midnight Commander также подходит для объединения файлов на компьютере моста. Экранный снимок на стр. 64 показывает *Midnight Commander*, работающий в оболочке *Bash* Windows 10, поставляемой с ОС. На правой панели отображаются текстовые каталоги, агрегированные на компьютере моста; обратите внимание, что файл семафора проверен в приведенной выше командной строке с *rsync*. На левой панели отображаются те же файлы, хранящиеся на внешнем терабайтном жестком USB-диске компьютера моста.

Восстановление утерянных данных

Важно знать, как восстанавливать сохраненные данные. Если *Rsnapshot* продолжает сохранять на компьютере моста, ответ тривиален. Просто просмотрите сохраненный том, выберите нужную дату и скопируйте нужные файлы.

Есть два способа восстановления файлов с удаленного компьютера резервного копирования. Один — вручную получить доступ к зашифрованным файлам на удаленном компьютере и просто скопировать файлы обратно в зашифрованный каталог компьютера моста, где они должны быть. Это можно сделать с помощью таких программ, как *Midnight Commander* или *ssh/rsync*. Вы можете получить файлы из каталога синхронизации на удаленном компьютере, или вам придется искать в каталоге **.stversions** на удаленном компьютере резервную версию файлов.

Основываясь на точном параметре вектора криптологической инициализации, который вы выбрали для *EncFS*, расшифровка файлов должным образом требует восстановления файлов в такие же зашифрованные структуры путей, что использовались первоначально — потому что с расширенными параметрами *EncFS* шифрование может зависеть от полного пути к файлу.

Чтобы получить текстовые копии файлов, необходимо также восстановить незашифрованный файл XML, который *EncFS* создала в зашифрованном каталоге компьютера моста. Если вы не сохранили отдельную копию XML-файла *EncFS*, убедитесь, что это один из файлов, которые вы получили извне сайта. Прежде чем *EncFS* декодирует файлы, вы должны иметь текстовую копию XML-файла для использования.

С помощью *Syncthing* можно также восстановить данные. Если **.stfolder** на синхронизируемом компьютере будет удалена, синхронизация остановится. Вы получите сообщение ошибки *synching Error on folder "ffff" (xxxxx-yyyuu): folder marker missing*. Если вы его использовали, также проверьте целостность другого файла *synching* — **.stignore**.

Можно вручную воссоздать **.stversion** и **.stignore**. Если вы используете дополнительный файл игнорирования, сначала воссоздайте **.stignore**, чтобы при возобновлении *Syncthing* не было утечки файла. Обратите внимание, что для создания файла **.stignore** вам надо будет использовать текстовый редактор, потому что опция графического интерфейса будет «заморожена», пока **.stfolder** отсутствует. Если **.stignore** исправен, для воссоздания **.stfolder** просто используйте *mkdir*.

С двумя исправными точка-файлами *synching* существует два способа восстановления файлов. Предположим, что вы случайно удалили или испортили кучу других файлов и ошибка распространилась на весь внешний компьютер, где по-прежнему есть хорошие копии в его каталоге **.stversions**.

Вот как восстановить зашифрованные версии. Используйте графические интерфейсы *synching*; только второй шаг потребует SSH-доступа к удаленному компьютеру или другу, которому вы можете позвонить по телефону, чтобы сделать удаленную копию.

1 Позвольте любым несвязанным файлам синхронизироваться с внешним местоположением, а затем приостановите компьютер моста внешнего синхронизирующего кольца и внешнего компьютера синхронизирующего кольца.

2 Скопируйте КОПИЮ файла или файлов из каталога **.stversions** на внешнем компьютере в исходное местоположение в структуре внешнего каталога. Если вы делаете отдельные файлы, то ниже смотрите описание, как найти эквивалентный зашифрованный путь и имя файла.

3 Измените синхронизирующее кольцо внешнего компьютера на *Send Only* и измените внешнее синхронизирующее кольцо компьютера моста на *Send/Receive*.

4 Возобновите действие обоих компьютеров и позвольте синхронизироваться файлам. Возможно, вам придется использовать кнопку перезаписи внешнего графического интерфейса, чтобы принудительно направить все изменения с внешнего сайта на компьютер моста.

5 Опять приостановите синхронизацию моста и внешнего сайта. Переключите мост обратно в *Send Only*, а внешний сайт обратно в *Send/Receive*. Снова запустите оба компьютера.

Быстрое переименование

Если в файле были последовательно сделаны два изменения и полностью размножены, управление версиями *trash-can* не сможет вернуться к «последней хорошей» версии. Однако, если вы используете резервные копии версии *Syncthing*, вы можете получить старые версии. *Syncthing* изменит зашифрованные имена файлов, добавив отметку даты, когда была сделана резервная копия.

Чтобы избежать ручного переименования всего зашифрованного файла после его восстановления и до того, как *EncFS* сможет их использовать, следующий скрипт может помочь, удалив последние 16 символов из имени файла. Для нашего урока мы скопировали его с форума пользователей *Syncthing*. См. форум помощи *Syncthing* для дальнейшего объяснения:

```
#!/bin/bash
for file in $(find /yourpath/ -type f -- regextype posix-awk --regex
"*[0-9]{8}-[0-9]{6}.*");
do
mv $file ${file::-16}
done
```

» ПЕРЕТАСУЕМ ФАЙЛЫ

Для дополнительной меры конфиденциальности вы можете смешивать различные наборы зашифрованных файлов в зашифрованном каталоге. Просто не забудьте убрать с дороги первый файл **.xml** (переименуйте его или переместите в другой каталог) перед созданием и использованием второго набора шифрования. За один раз можно использовать только один XML-файл.

Наложение разных перекрывающихся наборов шифрования в одну структуру каталогов делает зашифрованные файлы похожими на одну большую коллекцию, хотя перемешанные файлы зашифровываются разными файлами и разными паролями.

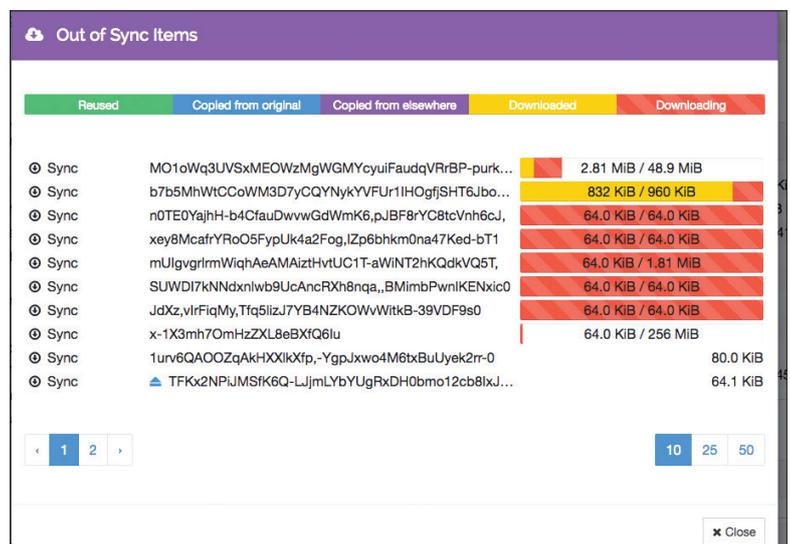
В текстовом каталоге отобразятся только файлы, соответствующие файлу XML и используемому паролю. Каждый созданный и используемый XML-файл позволяет дешифровать уникальное подмножество всех зашифрованных файлов. Например, вы можете распространять один большой набор шифрования и передавать разные файлы XML различным людям для доступа к определенным подмножествам файлов.

Команда `encfs showcrufft {encrypted-dir}` отобразит все файлы в зашифрованном каталоге, которые не могут быть декодированы, но без файла XML и соответствующего пароля все файлы будут отображаться как недопустимые. Это означает, что вы можете скрыть иголку в стоге сена — скажем, 10000 файлов, дешифрованных одним файлом XML, и два или три файла, дешифрованные вторым файлом XML и паролем, хранящимся где-то в другом месте. Никто не узнает, что второй файл XML был в игре, если они не знали паролей для декодирования одного или другого набора.

Поскольку *Midnight Commander*, или *rysnc*, или *Syncthing* вытаскивают файлы обратно на компьютер моста, *EncFS* преобразует зашифрованные файлы обратно в текст, и они будут синхронизированы с вашими рабочими компьютерами, если вы не приостановите текстовую синхронизацию колец.

Мы не знаем, почему *map*-страницы *EncFS* указывают на необходимость использования переменной среды `ENCFS6_CONFIG` для декодирования данных с внешнего компьютера («облако»). Это не обязательно, если вы помещаете XML-файл туда, где ожидалось — в реальный зашифрованный каталог данных. Однако вы можете поместить файл XML в любом месте и использовать переменную окружения:

```
$ ENCFS6_CONFIG=/somewhere_else/encfs6.xml encfs
~/recovery_files/encrypted ~/cleartextfiles. LXF
```



На экранном снимке показан графический интерфейс *Syncthing* внешнего сайта. Восемь файлов синхронизируются, остальные стоят в очереди.

DOCKER

Разверните свое первое виртуальное приложение

Трудно было убедить апатичного **Маянка Шарму** выключить его виртуальные машины и вместо этого научиться виртуализировать приложения.



НАШ ЭКСПЕРТ

Маянк Шарма — технический писатель, который проводит большую часть своего времени, играя в игры на Linux, как будто завтра не наступит.

Д аже если вы полностью отключены от юдоли смертных существ, вы всё равно наверняка слышали о *Docker* и о том, как он может решить все ваши ИТ-проблемы. Если вам удалось каким-то образом изолировать себя от благодати *Docker*, вот шанс искупить грехи ваши.

Традиционные технологии виртуализации обеспечивают полную аппаратную виртуализацию. Это означает, что виртуальная машина или гипервизор забирает часть физических ресурсов, таких как процессор, накопитель, ОЗУ, а затем разрезает их на виртуальные версии, такие как виртуальные процессоры и виртуальная оперативная память. Затем эти виртуальные периферийные устройства используются для создания виртуальных машин, которые ведут себя как обычные физические компьютеры. Изолированная виртуальная среда полезна для тестирования нового дистрибутива, но будет излишней, если вам надо виртуализовать одну программу.

И здесь контейнеры Linux предлагают через *Docker* привлекательную альтернативу. *Docker* позволяет связать любое приложение Linux со всеми его зависимостями и его собственной средой. Затем вы можете запускать несколько экземпляров приложения в контейнерах, причем каждый из них является полностью изолированным и отдельным процессом, с почти «родной» скоростью работы. Дело в том, что, в отличие от виртуальных машин, в контейнерах используется то же ядро главной системы. Это также означает, что на любом оборудовании вы можете размещать больше контейнеров, чем виртуальных машин, благодаря меньшим требованиям.

Язык Docker

Docker — это контейнерный механизм времени выполнения, позволяющий легко упаковывать приложения и помещать их в удаленный репозиторий, откуда другие пользователи могут их загружать и использовать. Давайте ознакомимся с терминологией *Docker*. Контейнеры *Docker* упаковывают программное обеспечение в полноценную файловую систему, которая включает всё необходимое для работы приложения. Это гарантирует, что приложение всегда будет работать одинаково — независимо от среды, в которой работает *Docker*.

```

bodhi@localhost:~/custom-ubuntu
File Edit View Search Terminal Help
[bodhi@localhost custom-ubuntu]$ sudo docker build -t custom_ubuntu .
Sending build context to Docker daemon 2.048kB
Step 1/5 : FROM ubuntu
----> 043804468dc
Step 2/5 : RUN sed -i 's/"#s"(deb."u"iverse))/\1/g' /etc/apt/sources.list
----> Running in cae39e109bd3
Removing intermediate container cae39e109bd3
----> e406758d095d
Step 3/5 : RUN apt-get -y update
----> Running in 3277a4e78c71
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Get:2 http://security.ubuntu.com/ubuntu xenial-security/universe Sources [73.2 kB]
Get:3 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [585 kB]
Get:4 http://archive.ubuntu.com/ubuntu xenial InRelease [247 kB]
Get:5 http://security.ubuntu.com/ubuntu xenial-security/restricted amd64 Packages [12.7 kB]
Get:6 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 Packages [405 kB]
Get:7 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 Packages [348 kB]
Get:8 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Get:9 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
Get:10 http://archive.ubuntu.com/ubuntu xenial/universe Sources [9802 kB]

```

Каждая команда в *Dockerfile* запускает новый контейнер и фиксирует новый слой образа перед переходом к следующей команде.

Образ *Docker* — это определение контейнера. Это набор всех необходимых исполняемых файлов, файлов, настроек среды и т.д., которые составляют приложение вместе с его зависимостями. Образ — это версия вашего приложения только для чтения, и его часто сравнивают с файлом ISO. Чтобы запустить этот образ, *Docker* создает из него контейнер, клонируя образ. Именно он затем на самом деле и выполняется. Такая компоновка делает *Docker* очень масштабируемым и позволяет запускать несколько контейнеров из одного и того же образа.

Хотя *Docker* доступен как пакет в официальных репозиториях всех популярных дистрибутивов, лучше всего получить последнюю версию из официального репозитория *Docker*. Запустите терминал и вызовите официальный скрипт загрузки; выполните его с помощью `curl -sSL https://get.docker.com/ | sh` для установки *Docker*. После установки запустите службу *Docker* командой `sudo systemctl start docker` и позаботьтесь, чтобы она запускалась при последующих загрузках, посредством `sudo systemctl enable docker`.

Теперь введите `docker run hello-world` для проверки установки. Команда загружает специальный образ из официального реестра *Docker*, который поприветствует вас, если всё будет

СКОРАЯ ПОМОЩЬ

Чтобы использовать *Docker* без полномочий `root`, добавьте своего пользователя в группу `docker` с помощью чего-то вроде `sudo usermod -aG docker bodhi`.

хорошо, и объяснит, какие шаги потребовались для проверки установки *Docker*.

Пойдем дальше и запустим новый контейнер *Docker*:

```
$ docker run -it --name alpha-silo ubuntu /bin/bash
```

С помощью этой команды мы попросили *Docker* запустить новый контейнер с образом под названием *Ubuntu*. `-i` делает сеанс интерактивным, а `-t` предоставляет терминал. Контейнер называется *alpha-silo* и запускает команду `/bin/bash`.

Когда мы даем команду, демон *Docker* будет искать образы *Ubuntu* в локальном кэше. Если образ не найден, он загружается из *Docker Hub*. Потребуется некоторое время, чтобы загрузить и извлечь все слои образов. *Docker* поддерживает образы контейнеров в виде нескольких слоев. В такой компоновке хорошо то, что эти слои можно разделять между несколькими образами контейнеров. Это делает систему очень эффективной. Например, если у вас на сервере запущена *Ubuntu* и вам надо загрузить контейнер *Apache* на основе *Ubuntu*, *Docker* загрузит только дополнительный слой для *Apache*, поскольку *Ubuntu* уже есть в локальном кэше, который можно использовать повторно.

Запустившись, этот контейнер выведет вас в новую оболочку, запущенную внутри него. Отсюда вы можете взаимодействовать с оболочкой так же, как и при обычной установке. Однако, поскольку контейнеры сконструированы с расчетом на чрезвычайную легкость, у вас есть доступ только к очень скудной среде.

Когда вы закончите, можно выйти из оболочки, набрав `exit` или нажав `Ctrl+D`. Вне контейнера вы можете использовать команду `docker ps` для вывода списка всех контейнеров и проверки состояния последнего контейнера. По умолчанию команда отображает только запущенные контейнеры. Добавьте в команду параметр `-a`, чтобы показать также и остановленные контейнеры. Чтобы снова запустить контейнер, можно использовать команду запуска *Docker*, например, `docker start -ia alpha-silo`.

Параметр `-i`, как и прежде, запускает контейнер в интерактивном режиме, а параметр `-a` прикрепит его к терминалу внутри контейнера. Если вы запускаете контейнер без каких-либо параметров, например, `docker start alpha-silo`, *Docker* запустит его в отключенном режиме, т. е. не зафиксирует контейнер на терминале, а просто запустит его в фоновом режиме.

Вы можете открыть терминал внутри отсоединенного контейнера с помощью `docker attach`, например, `docker attach alpha-silo`. Чтобы отсоединить терминал, но оставить контейнер в фоновом режиме, нажмите комбинацию клавиш `Ctrl+P+Q`. Чтобы выполнить команду внутри работающего контейнера, используйте `docker exec` — например, `docker exec alpha-silo pwd` выведет текущий рабочий каталог внутри контейнера.

Помните, мы сказали, что контейнеры предназначены быть легкими? Если вы отобразите все процессы, запущенные внутри нашего контейнера *Ubuntu*, с помощью команды `docker alpha-silo exec ps -elf`, вы заметите, что она запускает *Bash* и ничего больше. И когда мы выходим из оболочки, набрав `exit`, это заодно останавливает контейнер, поскольку это единственный процесс, выполняющийся в контейнере.

Команда `docker stop alpha-silo` изящно остановит контейнер после остановки процессов, запущенных внутри него. Когда контейнер вам больше не нужен, вы можете употребить `docker rm`, чтобы удалить или уничтожить его — например, `docker rm alpha-silo`. В таблице на стр. 70 перечислены некоторые часто используемые команды *Docker* и их назначение.

Наполним контейнер

Мы просто создали минимальный контейнер с именем *alpha-silo*, используя базовый образ *Ubuntu*, который мало что делает. Чтобы получить от этого контейнера больше пользы, вы можете загрузить еще один образ, который использует тот же базовый образ, но содержит больше начинки. Вы также можете вручную

```

bodhi@localhost:~
File Edit View Search Terminal Help
[bodhi@localhost ~]$ docker history custom ubuntu
IMAGE          CREATED          CREATED BY          SIZE
COMMENT
8f2aa8754a7a   8 hours ago     /bin/sh -c apt-get install -y build-essential 183MB
11e1974cc664   8 hours ago     /bin/sh -c apt-get -y upgrade                 32.2MB
1fcb9f71d50    8 hours ago     /bin/sh -c apt-get -y update                  40MB
4e33f9ce30b9   8 hours ago     /bin/sh -c sed -i 's/^#\s*(deb.*universe)\s... 2.76kB
0458a4468cbc   5 weeks ago     /bin/sh -c #(nop) CMD ["/bin/bash"]           0B
<missing>     5 weeks ago     /bin/sh -c mkdir -p /run/systemd && echo 'do... 7B
<missing>     5 weeks ago     /bin/sh -c sed -i 's/^#\s*(deb.*universe)\s... 2.76kB
<missing>     5 weeks ago     /bin/sh -c rm -rf /var/lib/apt/lists/*         0B
<missing>     5 weeks ago     /bin/sh -c set -xe && echo '#!/bin/sh' > /... 745B
<missing>     5 weeks ago     /bin/sh -c #(nop) ADD file:a3344b835ea6fcd56... 112MB
[bodhi@localhost ~]$
    
```

Список команд, применявшихся для создания образа, отображается командой `docker history` с именем образа.

добавить программное обеспечение к базовому образу, как при обычной установке. Запустите интерактивную оболочку внутри контейнера и введите

```
$ sudo apt update; apt install net-tools apache2 -y
```

Эта команда обновит репозитории и установит внутри контейнера *net-tools* и веб-сервер *Apache*. Одна из замечательных особенностей *Docker* заключается в том, что он позволяет сохранять свой настроенный контейнер в виде пользовательского образа, пригодного для развертывания дополнительных контейнеров.

Поэтому, если мы выйдем из контейнера и введем

```
$ docker commit -a "Mayank Sharma" alpha-silo loaded-silo
```

Docker свернет настроенный контейнер *alpha-silo* с обновленными репозиториями и веб-сервером *Apache* в пользовательский образ под названием *load-silo*. В команде параметр `-a` — это имя автора образа. Затем следует имя сворачиваемого в образ контейнера (*alpha-silo*), за которым следует имя нового образа (*load-silo*). Новый образ *load-silo* теперь сохраняется как отдельный образ на сервере вместе с другими, что можно проверить с помощью команды `docker images`. Теперь вы можете использовать этот образ для развертывания новых контейнеров.

СКОРАЯ ПОМОЩЬ

Для доступа к сетям вне локального сервера контейнеры маршрутизируют сетевой трафик через виртуальный интерфейс `docker0` на хост-сервере.

» DOCKER-ДИСТРИБУТИВЫ

Контейнер *Docker* включает как приложение, так и всю его конфигурацию и зависимости. Таким образом, вы можете оптимизировать дистрибутив чисто для размещения контейнеров. Сообщество зацепилось за это, так что у вас есть несколько урезанных контейнер-ориентированных дистрибутивов, например, *Container Linux*, *RancherOS* и *Atomic Host*. Хотя вы можете запускать контейнер поверх обычного дистрибутива, эти тонко настроенные дистрибутивы стали нормой для запуска контейнеров в производственной среде.

Каждый из этих дистрибутивов поставляется со своим набором функций, которые делают их пригодными для различных видов развертывания. *Container Linux* (ранее известный как *CoreOS*) — это готовая к производственной работе ОС, созданная с нуля для размещения контейнеров. Одно из ее преимуществ заключается в том, что она автоматически обнаруживает новый контейнер *Docker*, как только тот подключается к сети. Для управления контейнерами дистрибутив также использует *Google Kubernetes*. *A RancherOS* интересна тем, что по сути состоит из контейнеров *Docker*. Она загружается с помощью контейнера под названием *System Docker*, а затем дает пользователям возможность создавать новые контейнеры с помощью *User Docker*.

Если вы запускаете совместимые с *Red Hat* серверы с *CentOS*, *Fedora* или *RHEL*, то есть проект *Atomic Host*, который создает специализированные сборки для этих серверов *Red Hat* для интеграции *Docker* в вашу сеть. Есть также *Alpine Linux*, который начался как ветка проекта *LEAF* (*Linux Embedded Appliance Framework*). Его создатель теперь работает над *Docker*, который использует дистрибутив для сборки своих пакетов.



СКОРАЯ ПОМОЩЬ

Вы можете публиковать свои образы на Docker Hub, если у вас есть учетная запись в этом сервисе.

Практические контейнеры

Как мы уже говорили ранее, контейнер *Docker* является экземпляром образа *Docker*. *Docker* извлекает образы из репозитория, находящихся внутри реестров. Репозиторий *Docker* по умолчанию — это *Docker Hub*, где есть куча официальных и пользовательских неофициальных репозиториях, каждый из которых, в свою очередь, содержит множество образов.

Так что перейдите на <https://hub.docker.com>, чтобы просмотреть библиотеку готовых образов *Docker*. Чтобы познакомиться с *Docker*, мы используем его для установки приложения для блогов WordPress. Образ WordPress на *Docker Hub* не включает установку базы данных. Поэтому мы сначала должны установить базу данных *MariaDB* в отдельный контейнер, а затем попросить контейнер WordPress использовать его.

Начните с создания нового каталога, где вы хотите хранить файлы для *WordPress* и *MariaDB*, например, в своем домашнем каталоге:

```
$ mkdir ~/wordpress
$ cd ~/wordpress
```

Затем извлеките свежий образ *MariaDB*:

```
$ docker run -e MYSQL_ROOT_PASSWORD=<password> -e
MYSQL_DATABASE=wordpress_db --name db4wp -v $(pwd)/
database:/var/lib/mysql -d mariadb
```

Параметр `-e` задает переменные среды для контейнера, такие как пароль базы данных и его имя. Замените `<password>` своим паролем. Параметр `--name` определяет имя контейнера. Самым интересным параметром является `-v "$(pwd)/database":/var/lib/mysql`. Он просит *Docker* сопоставить два указанных местоположения, разделенных двоеточием (:). Справа находится каталог

`/var/lib/mysql`, который имеется в контейнере и используется для хранения файла базы данных. Команда просит *Docker* разместить файлы в папке `/database` в текущем рабочем каталоге на хосте, чтобы гарантировать сохранение данных даже после перезапуска контейнера. Параметр `-d` велит *Docker* запускать контейнер в режиме отдельного демона в фоновом режиме.

Эта команда загрузит последнюю версию официального образа *MariaDB* и поместит его в контейнер с заданными настройками. Вы можете убедиться, что контейнер *MariaDB* работает, с помощью `docker ps`.

Процесс также можно разбить на два этапа, что мы и сделаем для *WordPress*. Сначала мы просто загрузим образ *WordPress* с помощью `docker pull wordpress`, а затем создадим для него контейнер, с помощью

```
$ docker run -e WORDPRESS_DB_PASSWORD=<password> -d
--name my_wordpress --link db4wp:mysql -v $(pwd)/html:/var/
www/html -p <server public IP>:80:80 wordpress
```

Убедитесь, что вы задали переменной `-e WORDPRESS_DB_PASSWORD` тот же пароль, что и для базы данных *MariaDB*. Параметр `--link db4wp:mysql` связывает контейнер *WordPress* с контейнером *MariaDB*, чтобы приложения могли взаимодействовать друг с другом. Параметр `-v` выполняет ту же функцию, что и для базы данных, и гарантирует, что содержимое контейнера в каталоге `/var/www/html` постоянно сохраняется в папке `/html` в текущем каталоге на хосте.

`-p <server public IP>:80:80` велит *Docker* передавать соединения с HTTP-порта серверов на внутренний порт 80 контейнера. Замените `<server public IP>` публичным IP-адресом вашего сервера. Вместо публичного IP-адреса вы также можете использовать `-p 127.0.0.1:8080:80`, чтобы указать *Docker* о пробросе порта 80 в порт 8080 на локальном хосте. Чтобы получить доступ к установке *WordPress*, откройте браузер на компьютере в той же сети, что и сервер, на котором запущен демон *Docker*, и перейдите на <http://<IP-адрес сервера Docker>:8080>.

Команда `docker inspect wordpress` сообщит все настройки для контейнера *WordPress*. Чтобы проверить файл журнала для нашего контейнера *WordPress*, запустите команду `docker logs -f wordpress`. Вы можете остановить контейнер по `docker stop`, запустить его снова по `docker start` или перезапустить его по `docker restart`. Но если вам надо изменить параметр, например, сопоставление портов, сначала следует остановить контейнер, а потом удалить его и запустить другой с новыми параметрами, командой `docker run`.

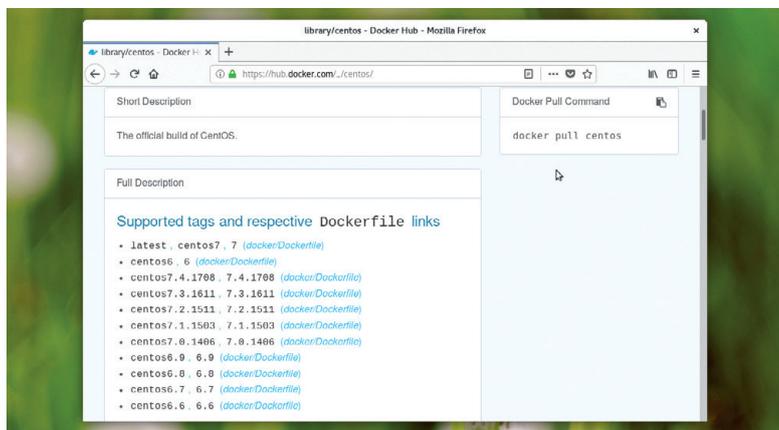
Docker Compose

Хотя интерфейс командной строки *Docker* хорошо документирован, это не самый интуитивный механизм для создания контейнеров. Вот почему вам нужно использовать инструмент *docker-compose* для определения и запуска контейнеров. Инструмент позволяет очень легко сворачивать несколько контейнеров. Он, по сути, состоит из человеко-читаемого языка сериализации данных YAML, перечисляющего характеристики или параметры одного или нескольких контейнеров, которые затем могут быть вызваны к жизни всего одной командой.

Чтобы продемонстрировать его преимущества перед CLI *Docker*, мы воссоздадим наши контейнеры *MariaDB* и *WordPress* с помощью *Docker Compose*. Сначала установите последнюю версию, вставив команду `cURL`, указанную в документации *Docker Compose* (<https://docs.docker.com/compose/install/#install-compose>). Загрузив и установив *Compose*, перейдите в папку `~/wordpress` и создайте файл `docker-compose.yml`:

```
$ cd wordpress
$ vi docker-compose.yml

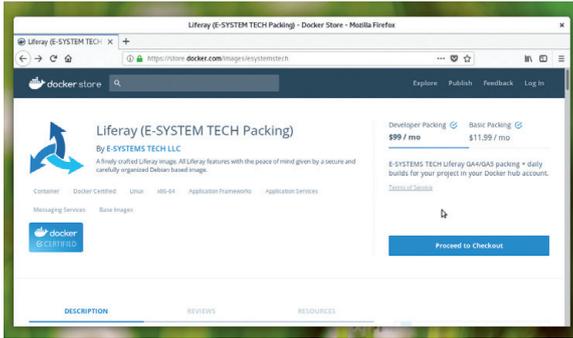
dbase4wp:
```



Для большинства образов, размещенных на Docker Hub, найдется любая информация, включая Dockerfiles.

» ТИПИЧНЫЕ КОМАНДЫ DOCKER

Команда	Что делает
<code>docker info</code>	Отображает сведения о демоне Docker
<code>docker ps</code>	Перечисляет все запущенные контейнеры
<code>docker inspect</code>	Отображает конфигурацию контейнера
<code>docker logs</code>	Отображает стандартный вывод контейнера
<code>docker exec</code>	Выполняет команду в запущенном контейнере
<code>docker image</code>	Отображает локальные образы
<code>docker run</code>	Иницирует новый контейнер
<code>docker pull/push</code>	Загружает/выгружает образы из/в удаленный реестр
<code>docker start/stop</code>	Запускает или останавливает контейнер
<code>docker rm</code>	Удаляет контейнер
<code>docker rmi</code>	Удаляет образ



Docker Store (store.docker.com) ориентирован на корпоративных пользователей и помимо бесплатных образов также содержит образы, размещенные на коммерческой основе.

```
image: mariadb
environment:
  MYSQL_ROOT_PASSWORD: <password>
  MYSQL_DATABASE: wordpress_db volumes: - ./database:/var/lib/mysql

my-wp:
  image: wordpress volumes: - ./html:/var/www/html
  ports:
    - "8080:80"
  links:
    - db4wp:mysql
  environment:
    WORDPRESS_DB_PASSWORD: <password>
```

Параметры те же, что и раньше, только подробнее. Сохраните файл, а затем введите `docker-compose up -d`, чтобы создать оба контейнера. Используйте `docker-compose logs -f` для контроля вывода контейнеров.

Создайте образ Docker

Как мы знаем, каждый контейнер *Docker* является экземпляром образа *Docker*. Конечно, в Docker Hub и в других местах есть огромное хранилище готовых образов *Docker*. Но так же, как мы ранее вручную наполнили образ Ubuntu, мы можем автоматизировать процесс и велеть *Docker* создать нам пользовательский образ, используя базовый образ.

Чтобы создать образ *Docker*, надо создать Dockerfile, который представляет собой текстовый файл с инструкциями и аргументами для сборки образа. В таблице вверху представлен список команд, которые входят в Dockerfile. Вам не нужно использовать каждую команду. Фактически, вот полностью функциональный Dockerfile:

```
$ vi Dockerfile
## Укажите базовый образ
FROM ubuntu:artful
## активируйте репозиторий Universe
RUN sed -i 's/^#\s*(deb.*universe)\$/1/g' /etc/apt/sources.list

## Обновите репозитории
RUN apt-get -y update

## Установите все доступные обновления
RUN apt-get -y upgrade

## Установите метапакет build-essential
RUN apt-get install -y build-essential
```

» КОМАНДЫ DOCKERFILE

Команда	Что делает
FROM	Указывает базовый образ
RUN	Используется для выполнения команды в процессе сборки
CMD	Выполняет команду внутри контейнера
WORKDIR	Задаёт путь, по которому должна выполняться команда, определенная с помощью CMD
ADD	Копирует файл с хост-компьютера в новый образ <i>Docker</i>
ENTRYPOINT	Команда по умолчанию, выполняемая при создании контейнера с образом
ENV	Определяет переменные среды
EXPOSE	Предоставляет определенный порт между контейнером и внешним миром
USER	Задаёт пользователя или UID для контейнера, созданного с образом
VOLUME	Разрешает контейнеру доступ к каталогу на хост-компьютере
MAINTAINER	Имя сопровождающего образа

Помните, однако, что, хотя вы можете разместить Dockerfile в каком угодно месте, когда вы создадите образ из него, любые файлы и каталоги в том месте или далее вниз по файловой системе в подкаталогах будут включены в сборку. Хорошая идея — создать каталог, специально предназначенный для размещения Dockerfile. Написав Dockerfile, вы можете употребить его для создания образа:

```
$ docker build -t custom_ubuntu
```

Эта команда создаст образ в текущем каталоге с именем **custom_ubuntu** на основе инструкций в Dockerfile. Когда это будет сделано, вы можете убедиться, что образ доступен вместе с другими образами, используя команду `docker images`. Теперь вы можете использовать этот пользовательский образ для создания контейнеров.

С помощью Dockerbuild можно сделать не только это. Фактически, наш урок затронул лишь малую часть *Docker*, но мы дали достаточно для начала. Если вы следовали за нами до конца, вы теперь должны быть обеспечены инструментами и ноу-хау, чтобы испытать совершенство и удобство контейнеров *Docker*. **LXF**



Существует несколько приложений с открытым исходным кодом, таких как *Rancher*, которые позволяют управлять развертыванием *Docker* через графический интерфейс типа «укажи и выбери».

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

Автоматизация сборки встроенных систем

Потренировавшись на сборке тулчейнов, **Дмитрий Кузнецов** перешел к сборке полноценного Linux.



НАШ
ЭКСПЕРТ

Дмитрий Кузнецов более 10 лет ведет неравный бой с монстрами сложности, хотя некоторые почему-то считают их ветряными мельницами.

В LXFD234 была напечатана статья «Анатомия тулчейнов». В ней рассказывалось о внутреннем устройстве GCC-тулчейнов и замечательной программе *Crosstool-NG*, автоматизирующей их сборку. В этой статье речь пойдет о *Buildroot* — еще одном инструменте из арсенала разработчиков ПО для встроенных систем. Он, как и *Crosstool-NG*, предназначен для автоматизации сборки, но не тулчейнов, а полноценных Linux-систем. Впрочем, собирать тулчейны *Buildroot* тоже умеет.

Устройство Linux-системы

Любая Linux-система состоит из трех компонентов:

- **Загрузчик.** Сразу после включения питания он получает управление, выполняет начальную инициализацию оборудования, ищет ядро Linux и передает ему управление. На настольных ПК обычно используется *Grub*, на встроенных системах популярен *Das U-Boot*.
- **Ядро Linux.** Оно завершает инициализацию устройств, принимает на себя управление ими, после чего ищет корневую файловую систему, монтирует ее в точку монтирования `/` и запускает первый процесс пользовательского режима (обычно `/sbin/init`).
- **Корневая файловая система (КФС).** Это основная часть системы. Технически она представляет собой обычную иерархически упорядоченную структуру файлов и каталогов, записанную на один из поддерживаемых Linux носителей информации (например, раздел жесткого диска) в формате одной из поддерживаемых Linux ФС (например, `ext4`). Особый статус КФС придает лишь роль, которую она играет в работе ОС. Эта роль заключается в обеспечении хранения всех установленных в системе программ и библиотек.

Сборка Linux-системы

Загрузчик и ядро Linux не зависят ни от каких дополнительных библиотек. Это значит, что, имея готовый тулчейн (а его легко получить с помощью *Crosstool-NG*), их нетрудно собрать вручную. С КФС дело обстоит не так просто. В современных Linux-системах могут быть установлены сотни, а то и тысячи неочевидно зависящих друг от друга пакетов ПО. Ясно, что сборка такой КФС — довольно трудоемкая задача. Иногда удается воспользоваться Linux-системами для настольных ПК (Ubuntu, Debian, Red Hat и т. д.), тем самым переложив проблему сборки ПО на плечи производителей дистрибутивов. Однако в мире встроенных систем такое решение не всегда оказывается приемлемым, поэтому

бывает, что даже «простым смертным» приходится разбираться во внутреннем устройстве КФС. К счастью, большая часть их содержимого — различные прикладные программы и библиотеки. Они хоть и полезны для решения определенных пользовательских задач, но не имеют никакого значения для поддержания работоспособности системы.

Действительно важных компонентов всего три:

- Набор системных программ и библиотек, которые собираются во время сборки тулчейна. *Crosstool-NG*, например, складывает их в отдельный подкаталог `sysroot`.
- Некоторые простейшие прикладные программы: *shell*, *mount* и т. д. Их можно получить одним махом, собрав “*busybox*” — огромное количество классических утилит Unix, реализованных в виде одного исполняемого файла.
- Минимальный набор стартовых скриптов и конфигурационных файлов в `/etc`. Для начала вполне хватит трех: *rcS*, *inittab* и *fstab*.

Такую минимальную КФС собрать довольно просто, однако она способна лишь на примитивную демонстрацию своей работоспособности. В реальных проектах обычно требуется нечто посложнее. Разумеется, несколько программ можно добавить и вручную, но если дело дойдет хотя бы до нескольких десятков, что вполне реально, стоит задуматься об автоматизации. Что же делать? Попробовать установить Ubuntu? Для встроенных систем такое решение часто оказывается слишком громоздким. А может и вообще не быть готового дистрибутива, собранного с нужными параметрами. В таких случаях на помощь приходит *Buildroot*.

Buildroot

Buildroot — это проект с открытым исходным кодом, предназначенный для автоматизации сборки Linux-систем. Конечно, основная сфера его применения — встроенные системы, ведь на настольном ПК гораздо проще и естественней установить какой-нибудь из известных дистрибутивов Linux (Ubuntu, к примеру). *Buildroot* может собрать всё: и тулчейн, и загрузчик, и ядро Linux, и КФС. Однако особый интерес представляет лишь последнее. Для сборки тулчейнов есть *Crosstool-NG* — более мощный и специально предназначенный для этого инструмент, а загрузчик и ядро Linux ненамного сложнее собрать и вручную.

Пользовательский интерфейс *Buildroot*, как и *Crosstool-NG*, построен на основе *Kbuild* — заимствованной из ядра Linux конфигурационной системы. Так что для опытного разработчика

```

dmntry@dmntry-M5-7519: ~/work/articles/buildroot/info
test -f /home/dntry/work/articles/buildroot/info/result/target/etc/ld.so.conf.d \
{ echo "ERROR: we shouldn't have a /etc/ld.so.conf file"; exit 1; } || true
test -d /home/dntry/work/articles/buildroot/info/result/target/etc/ld.so.conf.d \
{ echo "ERROR: we shouldn't have a /etc/ld.so.conf.d directory"; exit 1; } || true
mkdir -p /home/dntry/work/articles/buildroot/info/result/target/etc
  echo "NAME=Buildroot"; \
  echo "VERSION=2018.05.1"; \
  echo "ID=buildroot"; \
  echo "VERSION_ID=2018.05.1"; \
  echo "PRETTY_NAME=Buildroot 2018.05.1" "" \
} > /home/dntry/work/articles/buildroot/info/result/target/usr/lib/os-release
ln -sf /usr/lib/os-release /home/dntry/work/articles/buildroot/info/result/target/etc
>>> sanitizing paths in target tree
/home/dntry/work/articles/buildroot/info/buildroot-2018.05.1/support/scripts/fix-rpath target
touch /home/dntry/work/articles/buildroot/info/result/target/usr
>>> Generating common roots tarball
rm -rf /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs
mkdir -p /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs
cp -a /home/dntry/work/articles/buildroot/info/result/target /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/target
echo "#/bin/sh" > /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/fakeroot-fs
echo "set -e" >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/fakeroot-fs
echo "chown -h 0:0" >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/fakeroot-fs
printf "\n" >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/users_table.txt
PATH="/home/dntry/work/articles/buildroot/info/result/host/bin:/home/dntry/work/articles/buildroot/info/result/host/sbin:/home/dntry/bin:/home/dntry/local/bin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/dntry/work/projects/go/bin:/home/dntry/work/articles/buildroot/info/result/host/bin:/home/dntry/work/projects/go/bin" >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/fakeroot-fs
cat system/device_table.txt >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/device_table.txt
printf "\n\n" >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/device_table.txt
echo "#/bin/sh" >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/ext2/fakeroot
echo "set -e" >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/ext2/fakeroot
printf "\n\n" >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/ext2/target
tar -c -f /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/roots.common.tar --numeric-owner --exclude-THIS_IS_NOT_YOUR_ROOT_FILESYSTEM -C /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/target .\n >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/roots.common.tar
chmod a+x /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/fakeroot-fs
PATH="/home/dntry/work/articles/buildroot/info/result/host/bin:/home/dntry/work/articles/buildroot/info/result/host/sbin:/home/dntry/bin:/home/dntry/local/bin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/dntry/work/projects/go/bin:/home/dntry/work/articles/buildroot/info/result/host/bin:/home/dntry/work/projects/go/bin" >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/fakeroot-fs
rootdir="/home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/target"
table="/home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/device_table.txt"
>>> Generating root filesystem image roots.ext2
rm -rf /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/ext2
mkdir -p /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/ext2
echo "#/bin/sh" > /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/ext2/fakeroot
echo "set -e" >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/ext2/fakeroot
printf "\n\n" >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/ext2/target
tar -c -f /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/roots.common.tar --numeric-owner --exclude-THIS_IS_NOT_YOUR_ROOT_FILESYSTEM -C /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/target .\n >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/roots.common.tar
chmod a+x /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/ext2/fakeroot
PATH="/home/dntry/work/articles/buildroot/info/result/host/bin:/home/dntry/work/articles/buildroot/info/result/host/sbin:/home/dntry/bin:/home/dntry/local/bin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/dntry/work/projects/go/bin:/home/dntry/work/articles/buildroot/info/result/host/bin:/home/dntry/work/projects/go/bin" >> /home/dntry/work/articles/buildroot/info/result/build/buildroot-fs/ext2/fakeroot
mke2fs -l -l -L (24-Mar-2018)
creating regular file /home/dntry/work/articles/buildroot/info/result/images/roots.ext2
creating filesystem with 61440 1k blocks and 15360 inodes
filesystem UUID: F341e6c-676e-476d-926d-778aa99363c

```

Рис 1. Окончание сборки примера.

план действий сразу ясен: запустить конфигурационную систему (`make menuconfig`), выбрать нужные параметры, сохранить их в файл `.config` и запустить сборку (`make`).

Простейший пример

Очевидно, главная проблема при использовании *Buildroot* — разобораться с параметрами. Их много, а потому подробно описать каждый совершенно нереально, да и скучно было бы читать такой список. К счастью, в архив с исходным кодом *Buildroot* (подкаталог `configs`) разработчики включили множество примеров готовых конфигураций. Попробовать один из них гораздо эффективнее и интереснее. Затем, исследовав полученный результат, понять назначение конкретных конфигурационных параметров будет намного проще.

Скачаем с официального сайта последнюю версию *Buildroot* и распакуем полученный архив:

```
# wget https://buildroot.org/downloads/buildroot-2018.05.1.tar.bz2
# tar -xjvf ./buildroot-2018.05.1.tar.bz2
```

Buildroot готов к работе. Теперь можно переходить в каталог `./buildroot-2018.05.1` и приступать к настройке и сборке Linux-систем. Это самый простой путь, его часто предпочитают новички. Однако вместе с простотой он обладает и существенным недостатком — все результаты сборки останутся в дереве исходных кодов *Buildroot*. Для тех, кого такое решение не устраивает, есть чуть более сложный способ, позволяющий работать с *Buildroot* откуда угодно. Чтобы им воспользоваться, нужно каждой команде `make` передавать ключ `-C` с путем к дереву исходных кодов *Buildroot* и переменную окружения `O` с полным путем к каталогу, в который следует сохранить результаты сборки.

Допустим, исходный код *Buildroot* лежит в `./buildroot-2018.05.1`, а для хранения результатов будем использовать каталог `./result`. Его не обязательно создавать вручную, *Buildroot* отлично справится с этой задачей самостоятельно.

Посмотрим доступные примеры конфигураций:

```
# make -C ./buildroot-2018.05.1 O=$(pwd)/result list-defconfigs
```

Вполне подходящим вариантом кажется `qemu_arm_versatile_defconfig`. Попробуем:

```
# make -C ./buildroot-2018.05.1 O=$(pwd)/result qemu_arm_versatile_defconfig
```

Buildroot переключился на выбранную конфигурацию.

Можно посмотреть, а при необходимости и откорректировать ее параметры:

```
# make -C ./buildroot-2018.05.1 O=$(pwd)/result menuconfig
```

Выйдя из конфигурационной системы (клавишей `Esc`), запустим сборку:

```
# make -C ./buildroot-2018.05.1 O=$(pwd)/result
```

Если последняя команда завершилась без ошибок, *Buildroot* успешно справился с задачей (рис. 1).

Теперь полученную Linux-систему можно запустить на эмуляторе *QEMU* (имя пользователя — `root`, пароля нет):

```
# qemu-system-arm -M versatilepb -kernel result/images/zImage
-dtb result/images/versatile-pb.dtb -drive file=result/images/roots.ext2,if=scsi,format=ataw -append "root=/dev/sda console=ttyAMA0,115200" -serial stdio -net nic,model=r8139 -net user
```

Хотя ввод-вывод идет через терминал настольной системы, код действительно выполняется на ARM-эмуляторе.

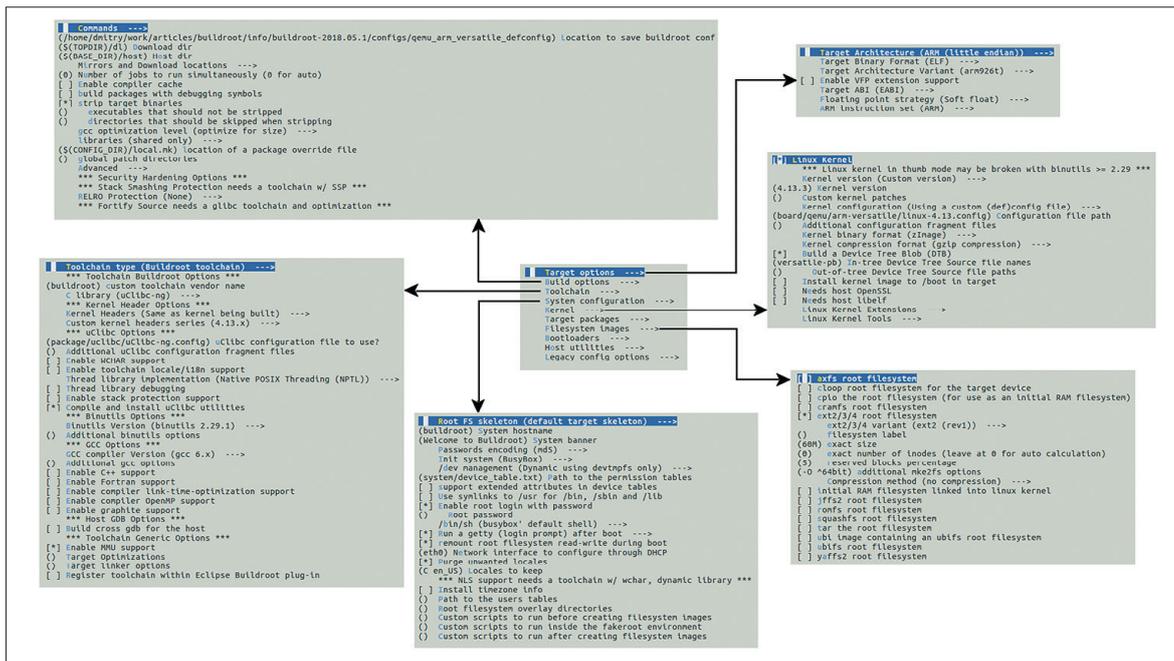
```

dmntry@dmntry-M5-7519: ~/work/articles/buildroot/info
pci bus 0000:00: root bus resource [mem 0x50000000-0x5fffffff]
pci bus 0000:00: root bus resource [mem 0x60000000-0x6fffffff] pref
PCI: BIOS: fast back to back transfers disabled
pci 0000:00:00: BAR 0: assigned [mem 0x00000000-0x00000fff] pref
pci 0000:00:01: BAR 2: assigned [mem 0x50000000-0x50001fff]
pci 0000:00:02: BAR 1: assigned [mem 0x50002000-0x500023ff]
pci 0000:00:03: BAR 0: assigned [io 0x1000-0x10ff]
pci 0000:00:0c: BAR 1: assigned [mem 0x50002400-0x500024ff]
pci 0000:00:0d: BAR 0: assigned [io 0x1400-0x14ff]
clcd-pl11x dev:20: Versatile hardware, VGA display
console: switching to colour frame buffer device 80x30
symb53c8x 0000:00:0d: enabling device (0100 -> 0103)
symb: -897as rev 0x0 at pci:0000:00:0d: irq 67
symb: No HWRAID, ID 7, Fast-40, LVD, parity checking
symb: SCSI BUS has been reset.
scsi host0: sym-2.2.3
Random: fast init done
symb: unknown interrupt(s) ignored, ISTAT=0x5 DSTAT=0x0 S1ST=0x0
scsi 0:0:0: Direct-Access QEMU HARDDISK 2.5+ PQ: 0 ANSI: 5
scsi target:0:0: Legged command queuing enabled, command queue depth 10.
scsi target:0:0: Beginning Domain Validation
scsi target:0:0: Domain Validation skipping write tests
scsi target:0:0: Ending Domain Validation
scsi 0:0:2: CD-ROM QEMU CD-ROM 2.5+ PQ: 0 ANSI: 5
scsi target:0:2: Legged command queuing enabled, command queue depth 10.
scsi target:0:2: Beginning Domain Validation
scsi target:0:2: Domain Validation skipping write tests
scsi target:0:2: Ending Domain Validation
libphy: Fixed MDIO Bus: probed
8139cp: 8139cp: 10/100 PCI Ethernet driver v1.3 (Mar 22, 2004)
8139cp 0000:00:0c: enabling device (0100 -> 0103)
8139cp 0000:00:0c: 0 eth0: RTL 8139cp at 0xc8025400, 52:54:00:12:34:56, IRQ 66
Versatile reboot driver registered
NET: Registered protocol family 10
sd 0:0:0: [sda] 122880 512-byte logical blocks: (62.9 MB/60.0 MiB)
sd 0:0:0: [sda] Write Protect is off
sd 0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
sd 0:0:0: [sda] Attached SCSI disk
Segment Routing with IPv6
sri: IPv6, IPv4 and MPLS over IPv4 tunneling driver
NET: Registered protocol family 17
Input: AT Raw Set 2 keyboard as /devices/platform/amba/amba:ppa/10006000.kni/serio0/input0
Input: INTxP2/ Generic Explorer Mouse as /devices/platform/amba/amba:ppa/10007000.kni/serio1/input0
EXT4-fs (sda): mount as ext3 due to feature incompatibilities
EXT4-fs (sda): mounting ext2 file system using the ext2 subsystem
EXT4-fs (sda): mounted filesystem without journal. Opts: (null)
VFS: Mounted root (ext2 filesystem) readonly on device 8:0.
devtmpfs: mounted
Freeing unused kernel memory: 132K
This architecture does not have kernel memory protection.
EXT4-fs (sda): warning: mounting unchecked fs, running e2fsck is recommended
EXT4-fs (sda): re mounted. Opts: block_validity,barrier,user_xattr
Starting Logging: OK
Initializing random number generator... done.
Starting network: 8139cp 0000:00:0c:0 eth0: link up, 100Mbps, full-duplex, lpa 0x05E1
udhcpd: started v1.28.4
udhcpd: sending discover
udhcpd: sending select for 10.0.2.15
udhcpd: lease of 10.0.2.15 obtained, lease time 86400
deleting routers
adding dns 10.0.2.3
OK
Welcome to Buildroot
buildroot login: root
# uname -m
armv5tej
#

```

Рис 2. Проверка на *QEMU*.

Рис 3. Конфигурация qemu_arm_versatile_defconfig.



Удостовериться в этом можно командой `uname -m`. Она должна вывести `armv5tej` (см. рис. 2).

Результаты сборки

Каталог для хранения результатов сборки задается переменной окружения `O`. В рассмотренном примере это `./result`. По умолчанию, если `O` не задана, **Buildroot** создаст в своем дереве исходных кодов каталог `output` и сохранит результаты туда.

В предыдущем разделе (в параметрах команды `qemu-system-arm`, запускающей собранную Linux-систему в эмуляторе **QEMU**) уже упоминалось содержимое подкаталога `./result/images`, но без пояснений. Теперь остановимся на нем более подробно:

- **zimage** — образ ядра Linux.
- **versatile-pb.dtb** — содержит описание устройств на плате ARM Versatile/PB. Это одна из широко известных плат на базе процессора ARM926EJ-S, эмуляция которой реализована в **QEMU**. Расширение DTB означает “device tree binary [англ. двоичное дерево устройств]”. То есть DTB-файлы — это двоичные образы иерархического представления аппаратных устройств вычислительной системы. Они не предназначены для чтения человеком, однако для каждого DTB-образа есть соответствующие DTS-файлы [англ. device tree source — исходник дерева устройств] исходного кода, написанные на специальном языке описания деревьев устройств. Преобразование из DTS в DTB выполняет компилятор `dtc` [англ. device tree compiler — компилятор деревьев устройств], который можно найти в каталоге `scripts/dtc` дерева исходного кода ядра Linux или установить из репозитория (в Ubuntu, например, так: `sudo apt-get install device-tree-compiler`). В исходном коде ядра Linux также можно найти **versatile-pb.dts (arch/arm/boot/dts)** и документацию по деревьям устройств (**Documentation/devicetree**).
- **rootfs.ext2** — образ КФС в формате ФС ext2. Подкаталог `./result/images`, вероятно, самый важный, однако другие тоже заслуживают внимания:
 - **./result/build** — исходный код и промежуточные результаты сборки собираемого ПО. Если сборка завершилась без ошибок, содержимое этого подкаталога вряд ли кого-то заинтересует, однако при решении проблем оно может быть полезным.
 - **./result/host** — вспомогательные инструменты, предназначенные для работы не на целевой машине (в данном случае ARM), а на машине, где выполняется сборка (в данном

случае — рабочий ПК x86_64). Хороший пример такого инструмента — кросстूलчейн.

- **./result/target** — незапакованное содержимое КФС. Однако стоит иметь в виду, что **Buildroot** при создании образа КФС выполняет над этими файлами ряд дополнительных операций (например, установку прав доступа, `setuid`).
- **./result/staging** можно назвать КФС для разработки. Здесь собран полный набор библиотек и заголовочных файлов включаемого в КФС ПО. Если какая-то программа имеет зависимости, то при ее сборке для целевой платформы в ключах `-I` и `-L` нужно передавать путь именно к этому каталогу. Зачем нужно было заводить отдельный каталог? Почему нельзя использовать КФС? Дело в том, что не всё содержимое **staging** попадает в КФС. Например, заголовочные файлы там совсем ни к чему. В мире встроенных систем целевые машины обычно слишком медленны и неудобны для сборки ПО, так что заголовочные файлы там будут только впустую занимать драгоценное место.

Параметры конфигурации

Теперь самое время заглянуть в параметры собранной Linux-системы. Они разбиты на десять разделов, которые можно увидеть на экране сразу после запуска конфигурационной системы командой:

```
# make -C ./buildroot-2018.05.1 O=$(pwd)/result menuconfig
```

Шесть наиболее важных показаны на рис. 3.

Target options [англ. параметры цели] объединяет параметры целевой машины:

- Процессорную архитектуру (ARM, MIPS, x86 и т.д.); в примере выбрана ARM little endian — ARM с обратным порядком байт.
- Конкретную модель процессора (в примере — arm926t).
- Формат двоичных файлов (ELF — используется на большинстве современных Linux-систем).
- Метод выполнения операций с плавающей запятой. Здесь можно выбрать программную обработку (Soft float) или использовать один из доступных на данной процессорной архитектуре аппаратных ускорителей. Для процессора arm926t таких ускорителей нет, однако если вместо arm926t взять, например, cortex-A15, они сразу станут доступны (VFPv2, VFPv3, VFPv4, NEON и т.д.).

Build options [англ. параметры сборки] задает общие параметры процесса сборки ПО. Они помогают **Buildroot** ответить на такие вопросы:

- С какими параметрами запускать необходимые при сборке команды (`wget`, `git`, `svn`, `tar` и т. д.)?
- Куда сохранить конфигурационный файл?
- Откуда скачивать архивы с исходным кодом?
- С каким уровнем оптимизации собирать ПО и нужно ли оставлять отладочные символы?
- Статические (*.a) или динамические (*.so) библиотеки следует включить в целевую КФС?

Для удобства задания путей в *Buildroot* предусмотрено ряд переменных. Наиболее часто используются `TOPDIR` и `BASE_DIR`. Первая содержит путь к дереву исходного кода *Buildroot*, вторая — каталог результатов сборки. Так, в примере через `TOPDIR` задан каталог для хранения архивов исходного кода: `$(TOPDIR)/dl`. Действительно, их можно найти в каталоге `./buildroot-2018.05.1/dl`.

Раздел `Toolchain` [англ. тулчейн] определяет параметры используемого тулчейна. Если в `Toolchain type` [англ. тип тулчейна] выбран вариант `Buildroot toolchain` [англ. тулчейн *Buildroot*], как в примере, *Buildroot* будет собирать тулчейн самостоятельно. Какие именно компоненты следует в него включить, уточняют остальные параметры этого раздела. На рис. 3 видно, что тулчейн будет построен на основе `GCC 6`, `binutils 2.29.1`, заголовочных файлов ядра Linux 4.13 и стандартной библиотеки `uClibc-ng`, с поддержкой `NTPL` [англ. Native POSIX Thread Library — библиотека «родных» потоков POSIX] и `MMU` [англ. Memory Management Unit — блок управления памятью], однако без компиляторов C++ и Фортрана.

Выбор в `Toolchain type` варианта `External toolchain` [англ. внешний тулчейн] позволяет использовать готовый тулчейн, например, собранный ранее с помощью *Crosstool-NG*. В этом случае нужно лишь указать путь, где его можно найти, и некоторые дополнительные параметры.

Чтобы получить работающую КФС, мало собрать необходимое ПО. Нужно еще выполнить целый ряд дополнительных действий. За их настройку отвечают параметры раздела `System configuration` [англ. конфигурация системы]. Их не так много:

- Задание временной зоны, локали, списка пользователей, имени хоста и пароля `root`.
- Выбор системы инициализации: `Busybox`, `SystemV`.
- Выбор метода создания файлов устройств в `/dev`: `devtmpfs` (метод, встроенный в ядро Linux v2.6.32+), `devtmpfs` вместе с `mdev` (специальная утилита из `Busybox`), `eudev` (форк от классического `udev`, разрабатывается сообществом Gentoo) или по старинке создание статического множества файлов устройств.
- Выбор командного интерпретатора и параметров `getty`.

Без этих параметров не обойтись при построении практически в любой Linux-системе. Что касается более экзотических возможностей, разработчики *Buildroot* вполне справедливо решили отказаться от запутанной и малопонятной для большинства пользователей системы настроек. Вместо этого они реализовали лишь несколько параметров, но настолько гибких и мощных, что при необходимости они позволяют сделать с КФС практически всё что угодно:

- `Root FS skeleton` [англ. основа КФС] задает каталог, содержимое которого копируется в КФС перед началом сборки. По умолчанию это `./buildroot-2018.05.1/system/skeleton` — набор пустых каталогов `dev`, `mnt`, `proc` и т. д., однако ничто не мешает указать в этом параметре что-нибудь другое.
- `Root filesystem overlay directories` [англ. каталоги, перекрывающие КФС] — задает набор каталогов, которые должны быть скопированы в КФС после окончания сборки.
- `Path to the permission tables` [англ. путь к таблице разрешений] — путь к списку файлов и каталогов КФС, для которых нужно вызвать `Busybox`-утилиту `makeperms`. По умолчанию — `./buildroot-2018.05.1/system/device-table.txt`.

```

/result/.config - Buildroot 2018.05.1 Configuration
Toolchain
> selects submenus ---> (or empty submenus --->). Highlighted letters are hotkeys. Pressing <Y> selects
for Search. Legend: [*] feature is selected [ ] feature is excluded

*** Toolchain External Options ***
Toolchain (Custom toolchain) --->
Toolchain origin (Pre-installed toolchain) --->
(/home/dntrty/x-tools/arm-unknown-linux-gnueabi) Toolchain path
(arm-unknown-linux-gnueabi) toolchain prefix
External toolchain gcc version (6.x) --->
External toolchain kernel headers series (4.10.x) --->
External toolchain C library (glibc/eglibc) --->
[*] toolchain has SSP support?
[*] toolchain has RPC support?
[*] toolchain has C++ support?
() Extra toolchain libraries to be copied to target
[*] Copy gdb server to the Target
*** Toolchain Generic Options ***
[ ] Copy gconv libraries
[*] enable MMU support
() Target Optimizations
() Target linker options
[ ] Register toolchain within Eclipse Buildroot plug-in

```

Кроме того, параметры `Custom scripts to run ...` позволяют передавать *Buildroot* произвольные сценарии, которые будут вызваны в определенные моменты сборки КФС.

Параметры `Filesystem images` [англ. образы ФС] дают возможность выбрать, в образы каких ФС необходимо запаковать полученную КФС по окончании сборки. В примере — `ext2`.

Разделы `Kernel` [англ. ядро] и `Bootloaders` [англ. загрузчики] определяют параметры сборки ядра Linux и начального загрузчика, если это необходимо.

Раздел `Target packages` [англ. пакеты цели] — огромный, разбитый по категориям список ПО, которое можно включить в КФС. Достаточно лишь выбрать необходимые элементы, а о загрузке исходного кода и сборке *Buildroot* позаботится самостоятельно.

`Host utilities` [англ. утилиты хоста] — аналогичный, но гораздо более скромный список вспомогательных утилит для хост-машины.

Раздел `Legacy config options` [англ. устаревшие конфигурационные параметры] заинтересует лишь тех, кто переходит с устаревших версий *Buildroot*.

Подключаем внешний тулчейн

В статье «Анатомия тулчейнов» (LXF234) было показано, как с помощью *Crosstool-NG* собрать полноценный кросс-тулчейн ARM/x86_64. Тогда его тестирование пришлось ограничить лишь парой примитивных примеров. Этого, очевидно, маловато для компилятора. Однако теперь, имея в распоряжении *Buildroot*, можно легко устроить проверку посерьезней. Для этого достаточно в уже рассмотренном примере `qemu_arm_versatile_defconfig` изменить настройки тулчейна, указав в `Toolchain type` вместо `Buildroot toolchain` — `External toolchain`. Ну и, конечно же, задать несколько довольно очевидных параметров, описывающих подключаемый внешний тулчейн. Как это сделать, показано на рис. 4.

Команды запуска сборки Linux-системы и ее проверки на эмуляторе *QEMU* уже были показаны выше (см. «Простейший пример»). Для данного случая они тоже годятся. Если набор ПО в собранной КФС кажется слишком аскетичным, то не составит труда ее пересобрать, предварительно запустив конфигурационную систему и выбрав там в разделе `Target packages` необходимые дополнительные программы и библиотеки.

Заключение

К сожалению, рамки журнальной статьи не позволяют поговорить о возможностях *Buildroot* более подробно. Одни вопросы оказались затронуты вскользь, другие вообще даже не упоминались. Самый важный и интересный из таких абсолютно обойденных вниманием вопросов — расширяемость. В конфигурационную систему *Buildroot* уже интегрировано более 2000 самых разных программ и библиотек, однако при необходимости этот список можно легко расширить. В документации *Buildroot* приведены подробные инструкции, как это сделать. LXF

Рис 4. Настройка внешнего тулчейна.

ALPINE + DOCKER

Плюс докеризация! Поваренная книга разработчика

Евгений Голышев рассуждает об основах создания Docker-образов и предлагает пар тройку полезных рецептов.



**НАШ
ЭКСПЕРТ**

Евгений Голышев не устает повторять, что разработчики без уверенных навыков владения Docker'ом — плохие разработчики. Признаться, уже достал этим всех..

Docker — одна из немногих технологий, которая, надев несколько лет назад много шума, до сих пор остается с нами. Чтобы помочь разработчикам взглянуть на любимый Docker под другим углом, я достал свою поваренную книгу, смахнул с нее пыль и приготовился к тому, чтобы в рамках этой статьи поделиться своими рецептами.

Два из четырех фигурирующих здесь рецептов призваны уменьшить размер результирующих образов. Эта задача всегда была и остается актуальной для Docker'a, поэтому я начну свой небольшой сборник рецептов с рассказа об Alpine Linux, а потом перейду непосредственно к самим рецептам. Во-первых, именно Alpine является первым шагом на пути к компактным образам, а во-вторых, я хочу воспользоваться случаем и рассказать немного об этом дистрибутиве, чтобы исправить недостаток информации об Alpine на страницах этого журнала (по какой-то причине журнал до сих пор обходил этот дистрибутив стороной).

Немного об Alpine

Alpine Linux применяется в качестве основы для официальных Docker-образов; в свое время он вытеснил с этого места Ubuntu. Более того, в начале 2016-го стало известно, что Натанаэль Копа [Nataanael Copra], создатель дистрибутива, присоединился к команде Docker'a. При этом Alpine остается независимым дистрибутивом, подконтрольным только своему сообществу.

Вот список наиболее интересных, на мой взгляд, фактов:

- Последняя на момент написания статьи версия дистрибутива — 3.8 — вышла 26 июня 2018 г., и ее официальный образ весит чуть больше 4 МБ.
- Разработчики стараются придерживаться предсказуемого графика выпуска новых версий дистрибутива. Как правило,

новые версии Alpine'a выходят два раза в год — в мае и декабре, но возможны и небольшие отклонения от графика. К примеру, версия 3.8 вышла с опозданием на пару месяцев.

- Каждая версия дистрибутива поддерживается на протяжении двух лет. Так как новые версии Alpine'a выходят два раза в год, то одновременно поддерживается 3-4 версии дистрибутива.
- В качестве системы инициализации используется *OpenRC*. Напомню, что стандартом де-факто сейчас является *systemd*.
- Поддерживается 6 портов: x86 и x86_64, armhf и aarch64, ppc64le, s390x. Это меньше, чем у Debian, но больше, чем у Ubuntu.

Статическая компоновка

Не вдаваясь в лишние подробности, сборку программы можно разделить на два этапа: получение объектных файлов (на каждую единицу компиляции — файл с расширением *.c* или *.cpp* — приходится по одному объектному файлу) и компоновку (или линковку, от *англ.* linkage). Суть компоновки заключается в создании исполняемого файла (или разделяемой библиотеки) из полученных ранее объектных файлов. Если у исполняемого файла есть зависимости в виде библиотек, то в задачи компоновщика также входит *связывание* [linking] исполняемого файла с этими библиотеками одним из двух способов: статически или динамически. Для статической компоновки необходимы *статические библиотеки* [static libraries], а для динамической — *разделяемые* [shared libraries].

Статическая библиотека представляет собой архив объектных файлов, который создается программой *ar*. Когда исполняемый файл компоуется со статической библиотекой, то составляющие данную библиотеку объектные файлы становятся частью этого

Докер-образы на базе текущей и предыдущей версий Alpine'a весят чуть больше 4 МБ.

```
eugulixes@ubuntu-mac: ~
File Edit View Search Terminal Help
eugulixes@ubuntu-mac:~$ docker images | egrep "^alpine"
alpine          3.8             11cd0b38bc3c   7 weeks ago   4.41MB
alpine          3.7             791c3e2ebfcb   7 weeks ago   4.2MB
eugulixes@ubuntu-mac:~$
```

исполняемого файла. Для сравнения, разделяемая библиотека представляет собой объектный файл. Когда исполняемый файл компонуется с разделяемой библиотекой, то в исполняемый файл попадает только информация об этой библиотеке. В таком случае, процесс компоновки будет осуществляться во время запуска программы динамически. (В силу ограниченного объема нашего урока, описание процесса компоновки пришлось сильно упростить, поэтому я настоятельно рекомендую обратиться к книгам «Linux API. Исчерпывающее руководство» Майкла Керриска и «Linux. Руководство программиста» Джона Фуско за более детальными подробностями по этому вопросу.)

С использованием разделяемых библиотек связан ряд преимуществ.

- Использование разделяемых библиотек уменьшает расход дискового пространства и памяти: на диске хранится и в память загружается только одна копия разделяемой библиотеки, независимо от количества программ, которые ее используют.
- Устранение ошибок и узависимостей в разделяемой библиотеке не требует пересборки зависимых от нее программ (конечно, при условии, что исправления не затронули двоичную совместимость).

Все эти преимущества не имеют никакой ценности в контейнерах. Во-первых, философия Docker гласит, что на один контейнер полагается только один процесс. Таким образом, в контейнере не должно оказаться двух процессов, которые бы разделяли код одной и той же библиотеки. Во-вторых, Docker-образы, как правило, обновляются атомарно — если один из компонентов контейнера требует обновления, то пересобирается весь образ.

Всё указывает на то, что статические библиотеки подходят для Docker-образов с идеологической точки зрения. Теперь осталось разобраться, есть ли в этом какая-то практическая польза.

Использование разделяемых библиотек негативно сказывается на размере Docker-образа. В основном это связано с особенностью дистрибутивов. В дистрибутивах разделяемые библиотеки распространяются в отдельных пакетах, а с установкой любых пакетов связаны определенные накладные расходы, и чем меньше остается пакетов в образе после его сборки, тем меньше он в итоге занимает места. Таким образом, если компактность результирующего образа для вас не пустой звук, то польза от использования статических библиотек определено есть.

К примеру, у Memcached есть две основные зависимости — пакет *libevent-dev* с библиотекой для асинхронного неблокирующего ввода/вывода, и *cyrus-sasl-dev* с библиотекой, реализующей вторую версию SASL API. *libevent-dev* зачем-то тащит за собой пакет *python2*, который в установленном виде занимает 39 МБ. Это именно те накладные расходы, о которых я говорил выше. Таким образом, официальный Docker-образ с Memcached'ом 1.5.10, последней на момент написания статьи версией демона, весит почти 59 МБ, т.к. постоянно таскает за собой Python 2, о котором даже никто не вспоминает, когда запускает Memcached. Docker-образ со статически скомпонованным с этими библиотеками Memcached'ом, весит чуть больше 10 МБ. Предлагаю в этом убедиться на примере образа Memcached'a из проекта MMB. Для этого сначала получите исходники всего проекта, а затем (предпочтительнее) соберите образ —

```
$ git clone https://github.com/tolstoyevsky/mmb.git
$ cd mmb/memcached
$ IMAGE_NAME=$(grep "image: " docker-compose.yml | awk -F: '{print $2}')
$ docker build -t ${IMAGE_NAME} .
```

или вытяните его с Docker Hub'a:

```
$ docker pull cusdeb/memcached:1.5.10-amd64
```

(На самом деле, в этом образе используется еще одна нехитрая оптимизация, о которой будет рассказано сразу в следующем рецепте. Без нее, конечно, размер образа был бы чуть больше.)

» GNU ИЛИ НЕ GNU/LINUX

Если очень сильно упростить, то любая Linux-подобная операционная система общего назначения состоит из следующих частей:

- утилит для работы с файлами, манипуляции текстом и управления процессами, которые составляют основу окружения любой Unix-подобной операционной системы;
- стандартной библиотеки языка C;
- ядра Linux.

Такие дистрибутивы, как Debian или Ubuntu (и многие другие), используют *GNU Coreutils* и *GNU C Library (Glibc)* в качестве Unix-подобного окружения и стандартной библиотеки языка C соответственно. Таким образом, Проект GNU, как разработчик этих компонентов, играет важную роль в дистрибуции, особенно если учесть тот факт, что для сборки вышеперечисленных частей используется компилятор C/C++ из состава коллекции компиляторов от GNU (*GNU Compiler Collection, GCC*). В связи с этим Debian и Ubuntu следует называть дистрибутивами GNU/Linux, т.е. дистрибутивами, в которых различные компоненты Unix-подобной операционной системы от GNU в сочетании с ядром Linux образуют прочную основу, на которой строится всё остальное.

Было даже время, когда полнофункциональную Linux-подобную операционную систему не представлялось возможным создать без привлечения арсенала GNU. Несмотря на то, что достойная альтернатива *Coreutils* в лице *Busybox* существовала с незапамятных времен, свободные компиляторы C/C++ и альтернативные стандартные библиотеки языка C не всегда дотягивали до функциональности своих GNU'шных собратьев. Но времена изменились, и сейчас *Musl* является достойной альтернативой *Glibc*, а *Clang* не только научился конкурировать с компиляторами C/C++ из состава GCC, но и начал в чем-то их превосходить. Таким образом, в новых операционных системах на базе ядра Linux появилась возможность без значительного ущерба для функциональности свести к минимуму использование кода от GNU, компенсировав его аналогами, распространяющимися под перmissive лицензией (Apache 2.0, все виды BSD, MIT и т.д.) или лучше «зачотченными» под встраиваемые устройства. К числу таких операционных систем принадлежит Alpine. (Однако, справедливости ради, стоит заметить, что значительная часть дистрибутива собирается не чем иным, как *GCC*.) Таким образом, в данном конкретном случае от GNU/Linux остается только Linux, в результате чего полное название дистрибутива — Alpine Linux. Данный подход является семантическим именованием: взглянув на одно лишь имя операционной системы, возможно не только понять, что она построена на базе ядра Linux, но еще и то, что в ней, к примеру, используется нечто отличное от *Glibc*. Для сравнения, полным названием Debian является Debian GNU/Linux. Глядя на это название, можно с уверенностью сказать, что концентрация распространяющегося под лицензией GPLv3 кода в базовой системе очень высока. К сожалению, не все проекты по разработке Linux-подобных операционных используют семантическое именование. Отличным примером является Gentoo Linux.

Но за эту оптимизацию приходится платить. Дело в том, что при использовании Alpine в качестве базового образа статическая компоновка осложняется тем, что у этого дистрибутива туговато со статическими библиотеками. К примеру, как *libevent-dev*, так и *cyrus-sasl-dev* содержат только разделяемые варианты библиотек. Наличие обоих вариантов библиотек в Alpine отдается на откуп сопровождающим этих библиотек, и если сопровождающий решил, что библиотека, от которой зависит докеризируемая вами программа, должна быть доступна только в разделяемом виде, то придется самостоятельно готовить ее статический вариант на этапе сборки самого образа, что потребует дополнительных сил написание Dockerfile'a.

Поэтому для начала необходимо решить, что для вас имеет больший приоритет — быстрая докеризация или небольшой объем результирующего образа. Если образ предназначен для широкого круга пользователей, то лично я бы вложился в уменьшение его объема.



```
eugulixes@ubuntu-mac: ~
File Edit View Search Terminal Help
eugulixes@ubuntu-mac:~$ docker images | egrep "(cusdeb/memcached|^memcached)"
cusdeb/memcached    1.5.10-amd64    5c9ca6f2ae7e    7 days ago    10.5MB
memcached          1.5.10         87d95d52bdd2    9 days ago    58.8MB
eugulixes@ubuntu-mac:~$
```

Образ со статически скомпонованным Memcached'ом весит почти в 6 раз меньше официального образа.

Докеризация только серверных частей

При докеризации клиент-серверного программного обеспечения не оставляйте в образе клиентов. В противном случае образ будет включать компоненты, которые примутся всюду за ним таскаться, и более того, о которых никто не вспомнит при запуске целевого сервера (немного напоминает историю с Python 2 и Memcached из предыдущего рецепта).

Клиенты, как правило, распространяются в дистрибутивах отдельно от серверов и, в отличие от последних, имеют минимум зависимостей и практически не требуют времени на их настройку, поэтому при решении узкоспециализированных я придерживаюсь идеи выкидывания из образа всего лишнего.

Тем не менее, если вам достаточно сделать `docker exec` для того, чтобы подключиться к серверу посредством клиента, запущенного в том же контейнере, что и сам сервер, то речь идет об учебной или просто несерьезной задаче, и волноваться вам не о чем. Во всех остальных случаях клиенты в Docker-образах с серверами являются лишними.

Сборка образов под ARM на машине x86

Благодаря широкому распространению одноплатных компьютеров (Raspberry Pi, Orange Pi, Banana Pi и т.д.), устройства на базе 32- и 64-битного процессора ARM, на которых можно запустить полноценную операционную систему, сейчас доступны как никогда. Более того, Docker поддерживает ARM уже достаточно давно: начиная с версии 1.10, вышедшей 4 февраля 2016 г., появилась возможность собрать клиентскую и серверную части под ARM, а с версией 1.12.1, вышедшей 18 августа того же года, официальная поддержка пришла на ARM-машины под управлением Debian Jessie, Ubuntu Trusty и Raspbian Jessie.

Тем не менее, Docker-образы не всегда удобно собирать непосредственно на одноплатниках — для экономии времени сборку лучше производить на рабочей станции или ноутбуке, которые, как правило, представляют собой машины на базе процессора x86. Один из способов этого добиться — поместить в *базовый образ* [base image] целевого Docker-образа *двоичные файлы средства эмуляции в режиме пользователя* [user mode emulation binaries].

Как в производных от Debian дистрибутивах, так и в Fedora двоичные файлы средства эмуляции в режиме пользователя содержатся в пакете `qemu-user-static`. Модуль ядра `binfmt_misc`, доступный в Linux начиная с версии 2.1.43 (которая, кстати, вышла в июне далекого теперь 1997 г.), позволяет распознавать различные форматы исполняемых файлов и ассоциировать их с произвольными приложениями. Другими словами, для определенного формата исполняемого файла можно зарегистрировать эмулятор, и при каждой попытке запустить исполняемый файл этого формата передавать его эмулятору, а не запускать на текущем процессоре. (В производных от Debian дистрибутивах также потребуется установить пакет `binfmt-support`, в который вынесена

функция регистрации эмуляторов.) Таким образом, если предназначенный для запуска на одноплатнике образ включает эти бинарники, то на машине на базе процессора, отличного от ARM, сборка образа и запуск контейнера на его основе будут осуществляться с привлечением эмулятора, а на ARM-машинах всё будет выполняться на реальном процессоре.

Предлагаю рассмотреть скрипт создания базового Docker-образа для ARM, который будет включать двоичные файлы средства эмуляции в режиме пользователя, чтобы любой образ на его основе мог быть без проблем собран и запущен на x86-машинах. Чтобы скрипт корректно отработал, в системе, где он будет запущен, должны находиться эти бинарники. В том случае, если этой системой является Fedora, установите пакет `qemu-user-static`, а если речь идет о Debian или Ubuntu, то `qemu-user-static` и `binfmt-support`.

А теперь перейдем к самому скрипту:

```
#!/bin/sh
set -e
if [ "$(id -u)" -ne "0" ]; then
  >&2 echo "This script must be run as root"
  exit 1
fi
mirror=http://dl-cdn.alpinelinux.org/alpine
alpine_ver=3.8
apk_ver=2.10.0-r3
chroot_dir=./alpine-baseimage
wget $(mirror)/v${alpine_ver}/main/armhf/apk-tools-static-${apk_ver}.apk
tar xzf apk-tools-static-${apk_ver}.apk
mkdir -p ${chroot_dir}/usr/bin
cp /usr/bin/QEmu-arm-static ${chroot_dir}/usr/bin
./sbin/apk.static -X $(mirror)/v${alpine_ver}/main -U --allow-untrusted --root ${chroot_dir} --initdb add alpine-base
mknod -m 666 ${chroot_dir}/dev/full c 1 7
mknod -m 666 ${chroot_dir}/dev/ptmx c 5 2
mknod -m 644 ${chroot_dir}/dev/random c 1 8
mknod -m 644 ${chroot_dir}/dev/urandom c 1 9
mknod -m 666 ${chroot_dir}/dev/zero c 1 5
mknod -m 666 ${chroot_dir}/dev/tty c 5 0
IMAGE=$(sh -c "tar -C ${chroot_dir} -c . | docker import -")
docker tag ${IMAGE} alpine:${alpine_ver}-armhf
```

Обратите внимание на то, что версия утилиты `apk-tools-static` (в данном случае 2.10.0-r3) является плавающей — почти наверняка к выходу данной статьи в свет она изменится. Скрипту определено следовало бы быть немного сложнее, чтобы учитывать эту особенность, но здесь он должен оставаться как можно более простым. Прежде чем переходить к его модернизации, проделайте следующее:

- откройте <http://dl-cdn.alpinelinux.org/alpine/v3.8/main/armhf/>, чтобы подсмотреть текущую версию `apk-tools-static`;
- обновите версию `apk-tools-static` в тексте скрипта;

```
eugulixes@ubuntu-mac: ~
File Edit View Search Terminal Help
eugulixes@ubuntu-mac:~$ docker images | grep "3.8-armhf"
alpine                               3.8-armhf                6ab54c337720            3 minutes ago            9.86MB
eugulixes@ubuntu-mac:~$ uname -m
x86_64
eugulixes@ubuntu-mac:~$ docker run --rm -it alpine:3.8-armhf sh
/ # uname -m
armv7l
/ #
```

● запустите скрипт с правами суперпользователя (т.к. скрипту необходимо производить различные манипуляции с chroot-окружением).

В результате будет создан базовый Docker-образ `alpine:3.8-armhf`.

Напоследок стоит сказать, что эмулятор позволяет решить подавляющее большинство задач, но, к сожалению, не всё через него запускается или собирается. Тем не менее, случаи отказов являются большой редкостью и сильно зависят от того, насколько экзотическими являются те вещи, для запуска которых эмулятор привлекается. По моему опыту, запущенные через эмулятор FTP-сервер `vsftpd` и сборка `watchtower` ведут себя нестабильно. К счастью, на этом список известных мне проблем заканчивается.

Накладывание патчей при сборке образа

Между пакетами исходных текстов, на базе которых строятся двоичные пакеты дистрибутивов GNU/Linux, и Docker-образами есть много общего. Нередки случаи, когда докеризируемая программа собирается из исходников. Как и в случае с пакетами исходных текстов, в рамках Docker-образа, возможно, придется заняться полировкой исходных текстов целевой программы — из-за того, что разработчики ее оригинала не учитывали особенности той системы, в которой программу в данный момент хотят заставить работать. Этот рецепт посвящен распространению патчей в составе исходников Docker-образов, чтобы, как и в случае пакетов исходных текстов, внесенные в тот или иной кусок (известного) кода изменения находились у всех на виду и могли быть проанализированы и переиспользованы другими разработчиками.

В качестве примера хочу привести Docker-образ `QEMU` из проекта MMB: <https://github.com/tolstoyevsky/mmb/tree/master/qemu>. Одной из главных задач для MMB в один прекрасный момент стало предоставление пользователям компактного образа с эмулятором (а по совместительству и системой виртуализации) `QEMU`, поэтому в качестве основы для образа был выбран Alpine. Затем, чтобы не зависеть от версии `QEMU`, которая в тот или иной момент времени находится в репозитории дистрибутива, было принято решение собирать эмулятор из исходников. После этого выяснилось, что некоторые зависимости `QEMU` отсутствуют в Alpine, и возникла необходимость их тоже собирать из исходных текстов.

Библиотека `numactl`, одна из зависимостей `QEMU`, до версии 2.0.12 не собиралась в Alpine из-за того, что `Musl`, стандартная библиотека языка C, которую использует дистрибутив, нигде не объявляет макрос `__GLIBC_PREREQ`, фигурирующий в одном из модулей библиотеки. Таким образом, `numactl` оказалась зависима от `Glibc`. Чтобы исправить эту ситуацию, был взят

на вооружение менеджер патчей `quilt`, тот самый, который используется в пакетах исходных текстов Debian.

К сожалению, `quilt` никак не может стать частью дистрибутива, вечно находясь в `edge`, тестируемой версии Alpine, поэтому менеджер патчей придется тоже собрать из исходников или написать примитивную альтернативу на базе программы `patch` и цикла `for`. Преимуществами использования `quilt` для управления патчами являются в первую очередь легкость отключения того или иного патча в серии или, при необходимости, изменение порядка их внесения.

Чтобы исправить сложившуюся с `numactl` ситуацию, можно воспользоваться самим `quilt`’ом, но для начала надо получить исходники библиотеки.

```
$ git clone https://github.com/numactl/numactl.git
$ cd numactl
```

Затем необходимо сообщить `quilt`’у о намерении создать новый патч:

```
$ quilt new adapt_to_musl.patch
$ quilt add syscall.c
```

`syscall.c` — тот самый модуль, в котором используется макрос `__GLIBC_PREREQ`. Теперь в него можно внести изменение, устраняющее проблему.

Моим решением на скорую руку было просто-напросто удалить `__GLIBC_PREREQ` из условия

```
#if defined(__GLIBC__) && __GLIBC_PREREQ(2, 11)
```

Это решает проблему и никак не влияет на логику — в случае использования `Musl` остаток выражения всегда будет ложным. Тем не менее, более красивым решением является добавление перед этим условием заглушки, которая выглядит следующим образом.

```
#ifndef __GLIBC_PREREQ
# define __GLIBC_PREREQ(x,y) 0
#endif
```

И наконец, надо обновить патч, чтобы он отражал сделанные только что изменения.

```
$ quilt refresh
```

В результате всех этих манипуляций в корне дерева исходных текстов библиотеки будет создана директория `patches` с двумя файлами — `adapt_to_musl.patch` и `series`. Второй файл является так называемой серией патчей и представляет собой список, который на данный момент состоит всего из одного-единственного патча. Посредством редактирования можно повлиять на поведение `quilt`.

Теперь, чтобы созданный патч был применен к `numactl` во время сборки образа, необходимо позаботиться о том, чтобы директория `patches` находилась в корне исходных текстов библиотеки. Затем, вызов `quilt push -a` иницирует накладывание всех патчей, указанных в серии. **LXF**

Возможность прозрачного запуска собранных под ARM программ на x86-машинах доступна в ядре более 20 лет..

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

HUGO

Создаем статический сайт с поддержкой тем

Михалис Цукалос применяет фреймворк Hugo для создания защищенных сайтов со статическими файлами. Вы и глазом не моргнете, а сайт уже готов!



НАШ
ЭКСПЕРТ

Михалис Цукалос — технический писатель, администратор UNIX, программист и математик. Он — автор книги «Системное программирование на Go», а сейчас пишет книгу «Освоение Go».

СКОРАЯ
ПОМОЩЬ

В Hugo вы можете создавать меню для своих сайтов. Подробнее см. на <https://gohugo.io/content-management/menus/>.

Go — популярный язык системного программирования, который можно использовать гораздо шире, чем работа с системным ПО. Один из примеров такого использования — Hugo. Согласно официальному описанию, Hugo — это настраиваемый генератор статических сайтов. С помощью этого фреймворка вы сможете создать сайт, состоящий из статических файлов, обойдясь без базы данных или других типов ресурсов, применяемых при создании динамических web-страниц.

Вперед, Golang

Итак, Hugo — это фреймворк, написанный на Go и позволяющий создавать сайты на основе статических файлов. Хотя Hugo написан на Go, для его употребления знать язык Go не нужно.

Учтите, что если вы ищете какую-то информацию о языке программирования Go, используйте ключевое слово “Golang” вместо “Go”, потому что “Go” — это обычное слово, и с ним вы будете завалены непригодной информацией. Проще говоря, используйте “Golang Hugo” вместо “Go Hugo”. На момент написания статьи последняя версия Hugo — 0.37.1, созданная на основе версии Go 1.10 (последняя версия Go).

Вы можете установить Hugo в Debian или Ubuntu Linux, выполнив первую из следующих трех команд:

```
$ sudo apt-get install hugo
```

```
$ which hugo
```

```
/usr/bin/hugo
```

```
$ ls -l /usr/bin/hugo
```

```
-rwxr-xr-x 1 root root 12980032 Feb 6 2016 /usr/bin/hugo
```

Две последние команды проверяют, что двоичный файл Hugo, с именем **hugo**, имеется в вашей системе. Однако, поскольку в вашем дистрибутиве Linux может не быть готового к установке пакета для Hugo или последней версии Hugo, возможно, вам потребуется зайти на сайт <https://gohugo.io/getting-started/installing> и установить Hugo вручную. Проще и быстрее всего загрузить Hugo из GitHub, как описано на предыдущей странице сайта, а не скачивать tar-архив.

После успешной установки Hugo вы можете просмотреть используемую версию Hugo, выполнив команду `hugo version`.

В следующем разделе мы объясним, как применить Hugo для разработки сайтов.

Действуем в Hugo

В Hugo необходимо выполнять определенные команды для создания нового проекта, для добавления в него одной или нескольких web-страниц, для загрузки и использования темы и т.д. Поэтому для создания структуры нового сайта вам понадобится выполнить команду `hugo new site` с именем структуры нового сайта. Затем выполните команду `hugo new` с другими аргументами, чтобы создать новую страницу или новую тему. Философия команд Hugo основана на способе работы компилятора Go.

```
2. mtsouk@ubu: ~/code/smallSite (ssh)
mtsouk@ubu:~/code$ cd smallSite/
mtsouk@ubu:~/code/smallSite$ ll
total 36
drwxrwxr-x 8 mtsouk mtsouk 4096 Mar 12 10:32 ../
drwxrwxr-x 2 mtsouk mtsouk 4096 Mar 12 10:32 ../
drwxrwxr-x 2 mtsouk mtsouk 4096 Mar 12 10:32 archetypes/
-rw-rw-r-- 1 mtsouk mtsouk 107 Mar 12 10:32 config.toml
drwxrwxr-x 2 mtsouk mtsouk 4096 Mar 12 10:32 content/
drwxrwxr-x 2 mtsouk mtsouk 4096 Mar 12 10:32 data/
drwxrwxr-x 2 mtsouk mtsouk 4096 Mar 12 10:32 layouts/
drwxrwxr-x 2 mtsouk mtsouk 4096 Mar 12 10:32 static/
drwxrwxr-x 2 mtsouk mtsouk 4096 Mar 12 10:32 themes/
mtsouk@ubu:~/code/smallSite$ hugo new posts/aWebPage.md
/home/mtsouk/code/smallSite/content/posts/aWebPage.md created
mtsouk@ubu:~/code/smallSite$ hugo new posts/pageNo2.md
Error: /home/mtsouk/code/smallSite/content/posts/aWebPage.md already exists
mtsouk@ubu:~/code/smallSite$ cat config.toml
baseurl = "http://replace-this-with-your-hugo-site.com/"
languageCode = "en-us"
title = "My New Hugo Site"
mtsouk@ubu:~/code/smallSite$ hugo new posts/pageNo2.md
/home/mtsouk/code/smallSite/content/posts/pageNo2.md created
mtsouk@ubu:~/code/smallSite$ hugo server -D

Your rendered home page is blank: /index.html is zero-length
* Did you specify a theme on the command-line or in your
  "config.toml" file? (Current theme: "")
* For more debugging information, run "hugo -v"

=====
2 of 2 drafts rendered
0 future content
2 pages created
0 non-page files copied
0 paginator pages created
0 tags created
0 categories created
```

На рисунке показаны структура сайта Hugo, добавление новых сообщений и запуск web-сервера Hugo для проверки работы сайта Hugo.

После создания некоторого базового содержимого вам потребуется изменить основной файл конфигурации сайта Hugo, чтобы изменить незатейливую версию сайта по умолчанию. Потребуется также изменить текстовые файлы на страницах созданного сайта, чтобы добавить данные на эти страницы.

По завершении настройки и добавления данных вам надо будет протестировать сайт, а затем развернуть его на внешнем web-сервере, где он станет доступным для вашей аудитории. При добавлении новых элементов на сайт Hugo потребуется снова загрузить его на сервер, чтобы изменения вступили в силу.

И последнее, но не по значению: перед тестированием сайта Hugo вам следует установить и активировать тему Hugo.

Небольшой сайт

В этом разделе мы расскажем, как создать простой сайт с именем smallSite с двумя web-страницами. Для начала выполним на нашем компьютере с Linux следующие команды:

```
$ hugo new site smallSite
$ cd smallSite/
$ hugo new posts/aWebPage.md
$ vi content/posts/aWebPage.md
```

Первая команда создает базовую структуру каталогов сайта Hugo, а третья добавляет на этот сайт новую web-страницу. Последняя команда, которая является необязательной, позволяет изменить содержимое созданной web-страницы. Заметьте, что команда `hugo new site` формирует удобный результат, содержащий полезную информацию для новичков в Hugo. После выполнения этих команд у вас появится структура каталогов, которая будет выглядеть следующим образом:

```
$ tree
.
├── archetypes
├── _default.md
├── config.toml
├── content
├── _posts
├── _aWebPage.md
├── data
├── layouts
├── static
├── themes
└── 7 directories, 3 files
```

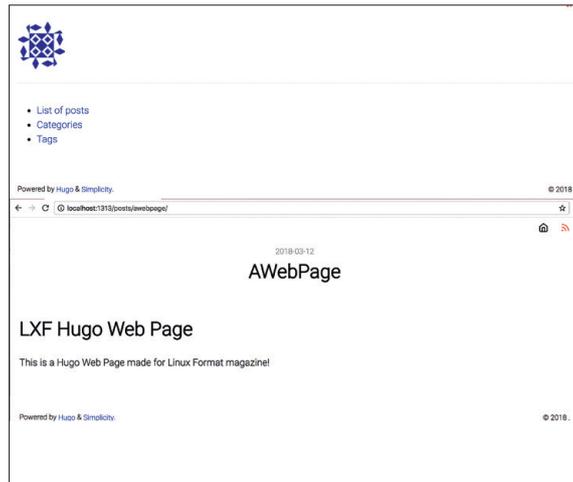
Теперь, когда вы ухватили смысл, создайте еще одну web-страницу с именем `posts/pageNo2.md`. Обратите внимание, что расширение файла `.md` означает Markdown. Как уже упоминалось ранее, если вы хотите изменить фактическое содержимое этих двух web-страниц, отредактируйте их в обычном текстовом редакторе.

Самый важный файл в каждом проекте Hugo называется `config.toml` и находится внутри корневого каталога проекта. Этот файл содержит важную информацию о конфигурации сайта Hugo. Его текущее содержание таково:

```
$ cat config.toml
baseUrl = "http://example.org/"
languageCode = "en-us"
title = "My New Hugo Site"
```

Учтите, что все строки в файле `config.toml`, которые начинаются с символа решетки (`#`), считаются комментариями. Первая строка определяет доменное имя сайта, а третья — название сайта. В Hugo можно тестировать сайты локально, на собственном сервере. Запускается этот сервер командой `hugo server -D` из корневого каталога вашего сайта Hugo.

В результате предыдущей команды будет указана информация о вашем сайте и HTTP-адрес web-сервера Hugo — по умолчанию Hugo прослушивает адрес `http://localhost:1313/` или



`http://127.0.0.1:1313/`. Заметьте, что эти два адреса эквивалентны, так как IP-адрес `localhost` в системах UNIX равен `127.0.0.1`. Однако, зайдя на любой из них, вы получите пустую страницу вместо сайта! Причина в том, что сейчас у сайта smallSite нет темы, поэтому он не может отображать страницы. В следующем разделе вы узнаете, как добавить тему на сайт Hugo!

На экранном снимке внизу стр. 80 показан результат выполнения некоторых команд, используемых в этом разделе.

Шаблоны

В Hugo для изменения внешнего вида сайта применяются шаблоны, которые в терминологии Hugo называются темами. Полный список тем Hugo см. на <https://themes.gohugo.io>.

Тема Hugo, которую будем использовать мы, называется `Simplicity`, и ее можно найти на странице <https://themes.gohugo.io/simplicity>. Чтобы загрузить и начать использовать тему `Simplicity`, выполните следующие команды:

```
$ git init; cd themes
$ git clone https://github.com/eshlox/simplicity.git
```

Вот две страницы проекта smallSite, созданные с помощью темы `Simplicity`. Разработчик сайта должен разместить данные на этих двух страницах.

СКОРАЯ ПОМОЩЬ

Дополнительную информацию о Hugo можно найти на странице <https://gohugo.io>, а дополнительную информацию о Go — на странице <https://golang.org/>. На сайте <https://themes.gohugo.io/> есть множество тем Hugo, а файлы разработки Hugo можно найти на странице <https://github.com/gohugoio/hugo>.

» ПЛЮСЫ И МИНУСЫ СТАТИЧЕСКИХ САЙТОВ

Основное преимущество сайта со статическими файлами — безопасность, поскольку взломать статический сайт, если web-сервер настроен правильно, очень сложно, если вообще возможно. Поэтому вам не придется беспокоиться о внедрении кода SQL или о легко подбираемых паролях! Второе преимущество статических сайтов — скорость, если сервер не перегружен. Кроме того, статические сайты легко индексируются, и их можно быстро перенести на другой web-сервер. Наконец, статические сайты гораздо дешевле в разработке, так как они разрабатываются гораздо быстрее.

Однако статические сайты имеют и свои недостатки. Самым большим недостатком статического сайта, созданного с использованием Hugo, является то, что даже при мелких изменениях надо каждый раз подгружать файлы на сервер. Второй недостаток статических сайтов в целом заключается в том, что их нельзя использовать с базой данных, а значит, нельзя и отображать web-страницы с динамическими данными. Кроме того, динамические сайты более эффективны по сравнению со статическими и предлагают полный спектр возможностей Интернета. Наконец, всегда есть шанс, что у вас получится статический сайт, который не соответствует первоначальным требованиям, а значит, вам придется создавать динамический сайт!

Несмотря на недостатки, если главное для вас — безопасность, стоит серьезно подумать о том, чтобы создать статический сайт — так вы получите более высокую скорость без дополнительных затрат.



СКОРАЯ ПОМОЩЬ

Два преимущества языка Go состоят в том, что он генерирует статические двоичные файлы и поддерживает кроссплатформенную компиляцию. Это означает, что у исполняемых файлов не будет зависимостей и вы можете создавать двоичные файлы для архитектур, отличающихся от текущей.

```
$ ls simplicity/
LICENSE i18n static
README.md images theme.toml
archetypes layouts webpack.config.js
data package.json yarn.lock
exampleSite src
$ ls -l simplicity/exampleSite/config.toml
-rw-r--r-- 1 mtsouk staff 1036 Mar 12 11:21 simplicity/exampleSite/
config.toml
```

Загрузив тему Simplicity, нам нужно указать эту тему в файле `./config.toml`. Если вы не можете запомнить, что надо сделать, чтобы использовать конкретную тему, есть сравнительно простое решение. Почти в каждой теме есть пример сайта с действительным файлом конфигурации `config.toml`. Для темы Simplicity этот файл `config.toml` находится в каталоге `simplicity/exampleSite/config.toml`.

Поэтому, чтобы использовать тему Simplicity, для сайта `smallSite` надо добавить в файл `./config.toml` следующую строку:

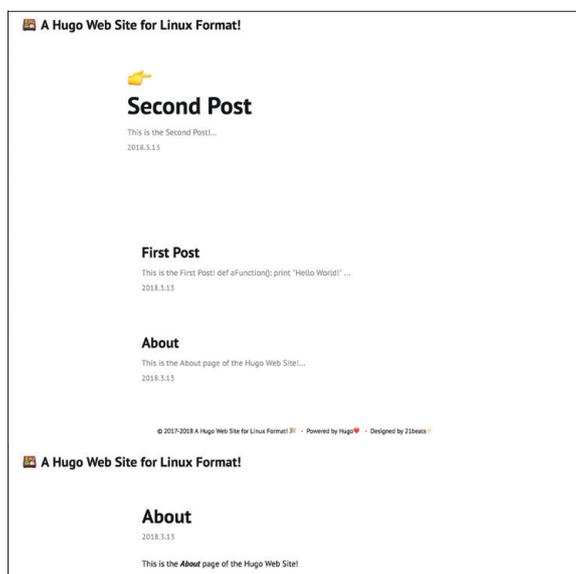
```
theme = "simplicity"
```

Экранный снимок на стр. 81 показывает две разные страницы сайта `smallSite` после применения темы Simplicity. Если вы попытаетесь использовать другую тему, например, `Osprey` (<https://themes.gohugo.io/osprey/>), предварительно ее не настроив, результат может оказаться не самым лучшим. Это означает, что некоторые темы Hugo лучше выглядят в варианте по умолчанию.

В Hugo, как и в Go, есть ряд строгих правил; так, если несколько раз использовать один и тот же ключ в файле `config.toml`, то при запуске web-сервера Hugo вы увидите следующее сообщение об ошибке:

```
$ hugo server -D
Error: While parsing config: (6, 1): The following key was defined twice: title
[Ошибка: При анализе config: (6, 1): Следующий ключ определялся дважды; имя]
rMacBook:smallSite mtsouk$ vi config.toml
```

Дополнительную информацию о параметрах команды `hugo server` можно получить, выполнив команду `hugo help server`. Параметр `-D`, использованный ранее, означает, что надо включить содержимое, помеченное как «черновик» [*англ.* `draft`]] — этот новый термин дает нам прекрасную возможность поговорить о различных типах содержимого Hugo.



На рисунке показана корневая страница (<http://localhost:1313/>) сайта блога, а также страница «О сайте» (<http://localhost:1313/about/>), предоставляемая web-сервером Hugo.

Но сначала поговорим о «вступлении [front matter]». В Hugo есть функция, позволяющая добавлять в файлы содержимого вступительную часть в формате YAML, TOML или JSON. Сейчас в файлах `aWebPage.md` и `pageNo2.md` используется формат YAML. Для формата YAML вступление — это всё, что находится между открывающими и закрывающими линиями `---`, как в следующем примере кода, взятого из файла `aWebPage.md`:

```
---
title: "AWebPage"
date: 2018-03-12T09:52:02+02:00
draft: true
---
```

Таким образом, вступление встроено в файл содержимого и поддерживает множество переменных, включая `draft`, `expiryDate` и `publishdate`. Если переменная `draft` в документе установлена в значение `true`, то этот документ не будет отображаться, если вы не запустите команду `hugo` с параметром `-D` или `--buildDrafts`. Переменная `expiryDate` определяет дату и время окончания публикации страницы. Наконец, переменная `publishdate` дает дату и время в будущем, в которое эта страница начнет публиковаться.

К сожалению, из-за ограниченного пространства нашего учебника мы не можем подробно описать оставшиеся переменные, используемые во вступлении. Полный список этих переменных можно найти на странице <https://gohugo.io/contentmanagement/front-matter>.

Настоящий сайт

На нашем следующем сайте будет страница `/about`, а также страница блога, которая по сути представляет собой основную страницу блога, где перечислены все доступные сообщения (посты) в блоге, чтобы посетитель сайта мог их прочесть. Почти во всех темах эта страница создается автоматически. На нашем сайте будет использоваться тема Hugo Paper, которую можно найти на <https://themes.gohugo.io/hugo-paper>.

Выполните следующие команды — они создают новый проект Hugo и загружают необходимую тему:

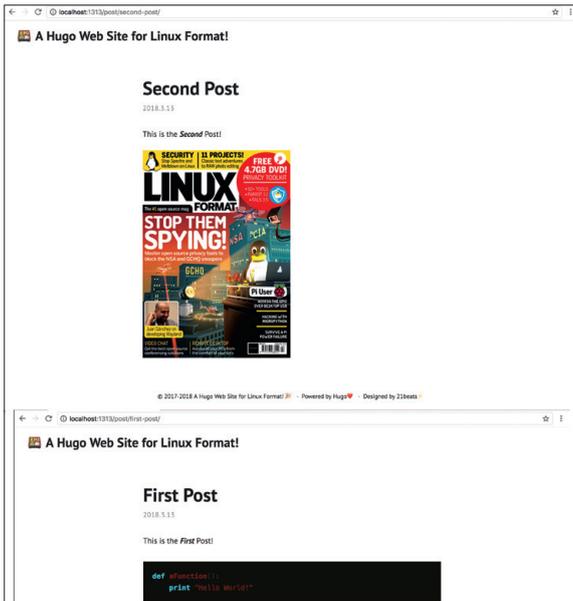
```
$ hugo new site blog; cd blog/themes
$ git clone https://github.com/nanxiaobei/hugo-paper
Затем создадим статическую страницу, а также две записи в блоге. Обратите внимание, что в одной из двух записей блога будет еще и рисунок:
$ hugo new about.md
$ hugo new post/First-Post.md
$ hugo new post/Second-Post.md
```

Окончательная версия файла `config.toml` такова:

```
baseURL = "http://hugo.mtsoukalos.eu/"
languageCode = "en-us"
title = "A Hugo Web Site for Linux Format!"
theme = "hugo-paper"
```

Теперь вы можете создать свой сайт, просто выполнив команду `hugo` или `hugo -D`. Лучше воспользуйтесь последней командой, так как две записи в блоге представляют собой черновики. Результаты команды `hugo -D` можно просмотреть в каталоге `./public`:

```
$ hugo
$ ls -l public/
total 288
-rw-r--r-- 1 mtsouk staff 2487 Mar 13 21:52 404.html
-rw----- 1 mtsouk staff 131063 Mar 13 21:41 LXF235.jpg
drwxr-xr-x 5 mtsouk staff 160 Mar 13 21:52 categories drwxr-xr-x 3
mtsouk staff 96 Mar 13 21:20 css
drwxr-xr-x 4 mtsouk staff 128 Mar 13 21:20 img
-rw-r--r-- 1 mtsouk staff 2452 Mar 13 21:52 index.html
-rw-r--r-- 1 mtsouk staff 516 Mar 13 21:52 index.xml
drwxr-xr-x 3 mtsouk staff 96 Mar 13 21:52 page
-rw-r--r-- 1 mtsouk staff 458 Mar 13 21:52 sitemap.xml
```



На рисунке показаны страницы <http://localhost:1313/post/first-post/> и <http://localhost:1313/post/second-post/>. Обратите внимание, что на первой странице отображается блок кода Python, а на второй странице – рисунок, который хранится в каталоге `static/` проекта Hugo.

```
drwxr-xr-x 5 mtsouk staff 160 Mar 13 21:52 tags
```

На экранном снимке стр. 82 показаны домашняя страница и страница «О сайте [about]», а на экранном снимке вверху — две записи в блоге. Запись `post/First-Post.md` также содержит код Python, который отформатирован и отображается в понятном виде.

Другие команды Hugo

Hugo поддерживает ряд дополнительных команд, кроме тех, которые уже были использованы в нашем учебнике. К ним относятся команды `benchmark`, `check`, `config`, `convert`, `env`, `gen`, `import` и `list`.

Команда `hugo benchmark` несколько раз создает сайт, анализируя процесс создания сайта, чтобы получить сравнительные показатели. Команда `hugo convert` используется для преобразования вступления в другой формат. Команда `hugo list` отображает содержимое сайта Hugo:

```
$ hugo list drafts
```

```
posts/pageNo2.md
```

```
posts/aWebPage.md
```

Информацию обо всех поддерживаемых командах и подкомандах Hugo см. на <https://gohugo.io/commands>.

Этап развертывания

На этом этапе вы помещаете файлы, созданные Hugo, на настоящий web-сервер с настоящим доменом, и этот сайт становится доступен вашей аудитории. Первое, что надо изменить перед развертыванием сайта Hugo с настоящим адресом на настоящем web-сервере — значение `baseUrl` в файле `config.toml`. На нашем уроке для развертывания сайта Hugo мы будем использовать поддомен <http://hugo.mtsoukalos.eu>.

Для настройки поддомена hugo.mtsoukalos.eu следует внести изменения на одном или нескольких DNS-серверах, а также создать учетную запись пользователя для хранения статических файлов сайта Hugo на сервере Linux. Однако эти действия опущены, так как зависят от конкретного хостинга. Таким образом,

» СОЗДАНИЕ НОВЫХ ТЕМ HUGO

Здесь мы поговорим о структуре проекта Hugo, для лучшего понимания, что происходит. В каталоге `archetypes` хранятся файлы для преднастроенных типов содержимого, которые будут использоваться командой `hugo new`. Использование этих файлов позволяет обеспечить согласованность между вашими web-страницами. Каталог `content` включает содержимое создаваемого сайта Hugo и может иметь внутреннюю структуру, отражающую различные типы содержимого.

Каталог `data` предназначен для хранения файлов настройки. В каталоге `layouts` хранятся файлы HTML, используемые Hugo для отображения содержимого вашего статического сайта. В каталоге `static` можно поместить любые статические файлы, включая CSS, JavaScript и файлы рисунков. В каталоге `public` Hugo помещает файлы, которые надо развернуть на web-сервере. Наконец, каталог `themes` предназначен для размещения ваших собственных тем или существующих тем с сайта www.github.com.

Hugo также позволяет создавать собственные темы; для этого выполните команду `hugo new theme` с названием темы. Темы Hugo следуют структуре сайта Hugo: файл макета темы по умолчанию находится в каталоге `layouts/_default/single.html`. Дополнительную информацию о различных параметрах команды `hugo new theme` можно получить, выполнив команду `hugo new theme`.

после настройки домена или поддомена и учетной записи пользователя UNIX, надо выполнить следующую команду Hugo, в которой нужно указать имя вашего web-сервера и учетную запись пользователя UNIX:

```
$ rsync -avz --delete public/ anAccount@machine:~/hugo
```

На экранном снимке внизу показан результат развертывания с использованием предыдущей команды, которая также отображает структуру каталога `public`. Не забудьте сначала выполнить команду `hugo` или `hugo -D`, чтобы развернуть последние версии статических web-страниц!

Вот и всё — теперь вы готовы приступить к разработке собственных безопасных и быстрых сайтов в Hugo! **LXF**

```
3. mtsouk@mail: ~ (bash)
iMac:blog mtsouk$ rsync -avz --delete public/ linode:~/hugo
building file list ... done
created directory /home/mtsouk/hugo
./
404.html
LXF235.jpg
index.html
index.xml
sitemap.xml
about/
about/index.html
categories/
categories/index.html
categories/index.xml
categories/page/
categories/page/1/
categories/page/1/index.html
css/
css/style.css
img/
img/apple-touch-icon.png
img/favicon.ico
page/
page/1/
page/1/index.html
post/
post/index.html
post/index.xml
post/first-post/
post/first-post/index.html
post/second-post/
post/second-post/index.html
tags/
tags/index.html
tags/index.xml
tags/page/
tags/page/1/
tags/page/1/index.html

sent 131116 bytes received 584 b
total size is 175607 speedup is
iMac:blog mtsouk$ vi content/post
iMac:blog mtsouk$ hugo -D
+-----+-----+
Pages | 12
Paginator pages | 0
Non-page files | 0
Static files | 4
Processed images | 0
Aliases | 4
Sitemaps | 1
Clenamed | 0
Total in 15 ms
iMac:blog mtsouk$ rsync -avz --de
building file list ... done
404.html
index.html
```

Здесь показано, как развернуть статические файлы сайта Hugo на удаленном сервере с помощью `rsync`. Обратите внимание, что `rsync` достаточно умен, чтобы обновлять только те файлы, которые были изменены.

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

WORDPRESS

Быстро создаем и клонируем сайт

Нужно создать клонированные сайты в нескольких подкаталогах с удобным управлением? **Кент Ельчук** проведет вас через все препятствия.



НАШ ЭКСПЕРТ

Кент Ельчук — web-разработчик и поклонник Linux, который в свободное время любит программировать и выращивать растения методом гидропоники.

На нашем уроке мы расскажем о такой возможности WordPress, как мультисайт. Для начала расскажем, что это такое. Мультисайт — это функция WordPress, позволяющая размещать несколько сайтов в одной установке WordPress с одной базой данных. Можно считать, что эти несколько сайтов расположены в поддоменах или подкаталогах. И мы можем войти на сайт, например, cat.example.com или dogs.example.com и управлять этими сайтами отдельно, как если бы это было несколько разных установок WordPress.

Применять мультисайт можно разными способами: представьте себе сайт с разными категориями и для нескольких муниципальных образований. Мультисайт позволит системному администратору сделать этот сайт уникальным для каждой категории или местоположения. Это не только удобно, но и полезно для SEO (поисковой оптимизации), так как родительский сайт может ссылаться на подсайты и быть связующим звеном для пользователей сайта из разных мест. При наличии мультисайта у вас будет главный администратор, который может зайти в основную консоль с установленными плагинами и запустить любой мультисайт. Главный администратор также может создавать новые мультисайты и создавать администраторов для любых мультисайтов.

Теперь кратко расскажем о клонировании. В WordPress есть несколько плагинов, с помощью которых можно клонировать любые сайты. Это позволяет легко создавать точные копии сайтов, передавая управление ими другим людям. Копия сайта будет включать весь исходный текст и картинки, но в нее можно вносить новые изменения, характерные именно для этой копии.

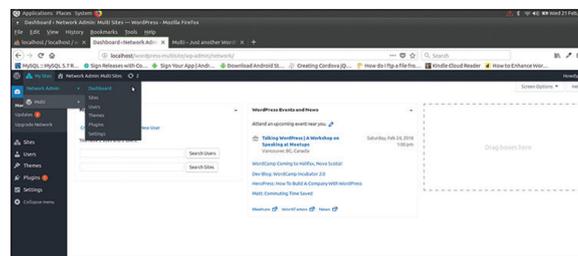
Осознав, как работают мультисайты и для чего они могут использоваться, перейдем к шагам по настройке мультисайта. Кстати, мультисайты можно применять не только к сайтам, создаваемым с нуля — да, мы можем взять готовый сайт WordPress и преобразовать его в мультисайт! Чтобы наш урок был полезен всем, мы опишем процесс установки с самого начала.

Создание нового мультисайта

Скачав WordPress и настроив базу данных, мы можем установить WordPress в базовой конфигурации. Затем откроем файл **wp-config.php** и внесем кое-какие мелкие изменения. Добавим код перед строкой `/* That's all, stop editing! Happy blogging. */`, чтобы получилось

```
/* Multisite */
define( 'WP_ALLOW_MULTISITE', true );
/* That's all, stop editing! Happy blogging. */
```

Затем в панели администрирования WordPress, доступной по адресу <http://localhost/wordpress-multisite/wp-admin>, откройте раздел Tools > Network Setup [Инструменты > Параметры сети]



С главной панели управления администратор сети может выполнять все действия по управлению, от создания новых копий до установки плагинов.

и нажмите Install [Установить]. После щелчка мышью вы увидите два фрагмента кода, которые можно добавить в файлы **.htaccess** и **wp-config.php** files. В корневом каталоге сервера Apache изменим содержимое файла **.htaccess** в соответствии с приведенным ниже фрагментом. Если этого файла нет, создайте его и добавьте следующие строки:

```
RewriteEngine On
RewriteBase /wordpress-multisite/
RewriteRule ^index\.php$ - [L]
# add a trailing slash to /wp-admin
RewriteRule ^([_0-9a-zA-Z]+)/wp-admin$
$1wp-admin/ [R=301,L]
RewriteCond %{REQUEST_FILENAME} -f [OR]
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^ - [L]
RewriteRule ^([_0-9a-zA-Z]+)/?(wp-(content|admin|includes).*) $2 [L]
RewriteRule ^([_0-9a-zA-Z]+)/?(.*)php$ $2 [L]
RewriteRule . index.php [L]
```

Далее приведен соответствующий код для файла **wp-config.php**:

```
define('MULTISITE', true);
define('SUBDOMAIN_INSTALL', false);
define('DOMAIN_CURRENT_SITE', 'localhost');
define('PATH_CURRENT_SITE', '/wordpressmultisite/');
define('SITE_ID_CURRENT_SITE', 1);
define('BLOG_ID_CURRENT_SITE', 1);
define('SUBDOMAIN_INSTALL', false);
define('DOMAIN_CURRENT_SITE', 'localhost');
define('PATH_CURRENT_SITE', '/wordpressmultisite/');
define('SITE_ID_CURRENT_SITE', 1);
define('BLOG_ID_CURRENT_SITE', 1);
```

Чтобы найти этот код в будущем, перейдите в раздел Network Admin > Settings > Network Setup [Администратор сети >

СКОРАЯ ПОМОЩЬ

С помощью `phpMyAdmin` или консоли `MySQL` создать резервную копию базы данных очень просто. Файлы легко упаковать и сохранить в безопасном месте.

> Параметры > Настройка сети]. Итак, у нас есть рабочий мульти-сайт, и мы можем создавать новые сайты. Наш исходный сайт находится по ссылке `localhost/wordpress-multisite`. Выбрав Sites [Сайты] из меню слева, мы увидим этот URL-адрес.

Теперь давайте создадим новый сайт — чтобы разобраться, как это работает. Для этого достаточно нажать кнопку Add New [Добавить новый сайт] вверху и указать URL-адрес, название сайта и адрес электронной почты. Затем нажмите кнопку Add site [Добавить сайт], и у вас появится новый сайт.

Вернувшись к списку сайтов, мы увидим, что там появился адрес `localhost/wordpress-multisite/new`. Наведя указатель мыши на список и нажав Visit [Перейти], мы увидим, что сайт выглядит точно так же, как оригинал. Нажав Dashboard [Панель управления], мы сможем управлять сайтом. Если вы растерялись и не знаете, что делать дальше, выберите Network Admin [Администратор сети] или Network > Admin > Dashboard [Сеть > Администратор > Панель управления] рядом со значком WordPress в левом верхнем углу. Затем можете выбрать Sites [Сайты] и перейти к любому из существующих сайтов.

При создании таким способом новых сайтов к каждому сайту применяются простые настройки по умолчанию. И если изменить исходный сайт и вложенные сайты, на главной странице сайта всё равно будет находиться запись “Hello World”. Во многих случаях это как раз и нужно. В других случаях можно создать копию уже созданного сайта. Для этого надо установить плагин для клонирования сайтов.

Приглашаем клонов

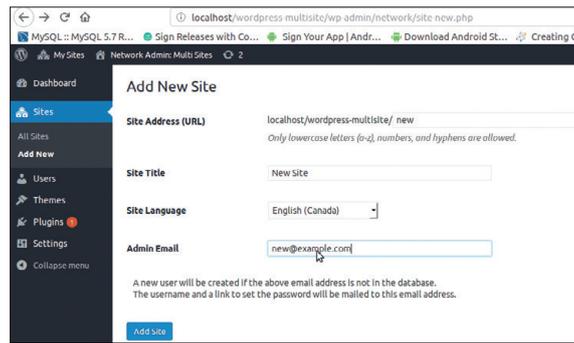
Чтобы установить плагин, перейдите на панель управления Network Admin [Администратор сети], наведите курсор мыши на Plugins [Плагины] и выберите Add New [Добавить]. В правой части страницы можно искать плагины. Введите “cloner” или “NS cloner” в строке поиска, и вы обнаружите плагин NS Cloner.

После этого нажмите кнопку Install Now [Установить сейчас]. Для установки плагина вручную его можно скачать с сайта <https://en-ca.wordpress.org/plugins/ns-clonersite-copier>. Скачав плагин, распакуйте файл и переместите папку плагина в папку `wp-content/plugins`. После установки (вручную или через сайт) плагин появится в списке установленных плагинов на панели управления Network Admin [Администратор сети]. Его можно найти в разделе Plugins > Installed Plugins [Плагины > Установленные плагины]. Затем при необходимости выберите Network Activate [Активация по сети]. После активации на панели управления Network Admin появится новый плагин, в названии которого будет “NS Cloner”. На момент написания статьи это NS Cloner V3.

Давайте, не откладывая, попробуем плагин в действии. Для начала просто нажмите на NS Cloner V3. Вы сможете увидеть весь текст и все рисунки, связанные с плагином. Чтобы перейти прямо к делу, выберите один из существующих сайтов в поле Select Source [Выбрать источник]. Это может быть как исходный сайт, так и один из сайтов, созданных с помощью мультисайта.

Теперь надо добавить заголовок. Для простоты назовем этот сайт First Clone [Первый клон]. Ниже нужно указать URL-адрес. Пусть это будет `first-clone`. Затем нажмите кнопку Clone [Клонировать]. После нажатия кнопки Clone сайт копируется достаточно быстро. Однако с настройками PHP по умолчанию скорее всего появится предупреждение о превышении параметра `max_execution_time`, который по умолчанию равен 60 секундам. Кроме того, плагин может сообщать о некоторых ошибках при записи в лог-файл.

Даже с этими предупреждениями клонирование скорее всего завершится успешно. Но если вы хотите, чтобы предупреждение о превышении `max_execution_time` не появлялось, можно создать простой файл PHP и вывести информацию о PHP. Это делается одной командой:



```
<?php echo phpinfo();
```

Открыв эту простенькую страницу в браузере, найдите параметр Loaded Configuration File [Загруженный файл конфигурации] в столбце слева. Это примерно пятая по счету строка сверху. Затем откройте этот файл в редакторе, например *vim* или *nano*, найдите параметр `max_execution_time` и измените его значение на 60. Мы пользовались этим плагином в нескольких случаях, даже для сайтов с большим количеством рисунков, страниц и записей, и для клонирования обычно требовалось всего несколько секунд, гораздо меньше, чем 30 секунд. Так что насчет предупреждения `max_execution_time` сильно не волнуйтесь.

При этом помните, что при создании клона будут скопированы весь текст и все изображения исходного сайта. Что касается проблемы с файлом журнала, надо просто изменить права доступа к папке `wp-content/plugins/ns-cloner-site-copier/logs`. Для этого в командной строке перейдите в эту папку и измените права доступа. Команды приведены ниже:

```
cd /var/www/html/wordpress-multisite/wp-content/plugins/  
ns-cloner-site-copier  
chmod -R 777 logs
```

Ну вот, мы умеем клонировать и создавать новые сайты без клонирования; давайте заглянем внутрь и разберемся, что происходит. По сути, это изменения файлов и папок в каталоге `wp-content/uploads`. Если на локальном компьютере Linux нет этого каталога, его можно создать и назначить для него разрешения 777. Это делается следующими командами:

```
$ cd /var/www/html/wordpress-multisite/wp-content
```

» УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ

Одним из основных преимуществ мультисайтов в WordPress является возможность создавать подсайты, которыми могут управлять другие люди. Таким образом, наш мультисайт может представлять собой несколько более мелких сайтов, обслуживаемых людьми из разных мест. В качестве одного из примеров обслуживания сайта местными пользователями можно привести сайт о местных мероприятиях, например, турпоходах. Каждым из подсайтов будет управлять пользователь из соответствующего местоположения, что позволит сделать этот подсайт информативным для его соседей.

Для реализации такой схемы надо назначить администраторов для каждого из наших сайтов и отдать им на откуп всё остальное. Клонировем новый сайт, назовем его “clone” и установим для него следующий URL-адрес: <http://localhost/wordpressmultisite/clone>. Теперь любой из пользователей этого сайта может открыть страницу <http://localhost/wordpress-multisite/clone/wp-admin> и войти на нее с соответствующими привилегиями.

При использовании мультисайта с несколькими администраторами для каждого подсайта управлять всеми сайтами, включая новые сайты и клоны, может администратор сети для исходной учетной записи. При этом любые привилегии на каждом отдельном клоне или сайте будут распространяться только на данный клон или сайт.

При создании нового сайта будет создан пустой сайт в новой подпапке. Простенький, но для начала сгодится.



```
$ mkdir uploads  
$ chmod 777 uploads
```

При добавлении изображений на любой из наших сайтов эти изображения будут находиться в одной из папок в каталоге **Uploads**. Изображения для исходного сайта будут находиться в папке **uploads/2018/04**. 2018 и 04 — это текущий год и текущий месяц. Изображения для новых и клонированных сайтов будут находиться в каталоге **uploads/sites/2/2018/04**. 2 — это номер клона в последовательности. Например, клон с путем **uploads/sites/3/2018/04** был создан после клона с путем **uploads/sites/2**.

Настройка таблицы

В отличие от изображений, новые таблицы в базе данных создаются при каждом клонировании сайта или при создании сайта без клонирования. Рассмотрим, как это работает. У таблиц для исходного сайта будет префикс **wp_**, например, **wp_options**.

При создании новых сайтов для них создаются новые скопированные таблицы с другим префиксом, например, **wp_2_options**. Номер для каждого нового сайта будет увеличиваться на единицу. Эта нумерация базы данных позволяет легко сопоставлять файлы в папке **Uploads** с базой данных, на случай, если пользователи захотят внести изменения на уровне базы данных. Кроме того, нам надо знать, какие папки и таблицы базы данных удалять, если мы захотим удалить старые сайты, поскольку при удалении сайта на панели управления Network Admin [Администратор сети] эти папки и таблицы не исчезнут.

Итак, у нас есть несколько копий и клонов; вернемся немного назад и посмотрим, как управлять сайтами. Мы знаем, как получить доступ к панели управления сайта. Теперь рассмотрим еще несколько ссылок в списке Sites [Сайты] на панели управления Network Admin [Администратор сети]. Эти ссылки помогут нам выполнять другие действия, не переходя на панель управления. Например, мы можем деактивировать сайт, архивировать сайт, пометить сайт как спам, удалить сайт и перейти на сайт. При наличии плагина NS Cloner мы также можем клонировать сайт.

Хотя у нас есть довольно большой список действий, основным инструментом, которым мы будем пользоваться, будет панель управления, так как она позволяет внести изменения в отдельный сайт. Если отключить сайт, он не будет отображаться в браузере, а если удалить сайт, он исчезнет. Однако стоит помнить, что отключенный сайт всё равно будет отображаться в браузере, если войти в него в режиме администратора; но его нельзя будет просмотреть в другом браузере, для которого вы не авторизовались.

Поиск и замена

При клонировании сайта с помощью NS Cloner нам скорее всего надо будет изменить в копии текст и изображения. Изменить

изображения очень просто. Достаточно загрузить новые изображения и удалить ненужные изображения на панели управления сайта с помощью ссылки Media [Мультимедиа] в главном меню слева. Но с текстом всё иначе.

Давайте припомним идею о кошках и собаках, о которой мы говорили ранее. Предположим, что мы хотим изменить весь текст созданного клона и заменить слово “dog” на слово “cat” — во всех заголовках и во всем тексте на различных страницах и в различных записях. Это сэкономит нам драгоценное время.

Для начала надо перейти в панель управления Network Admin [Администратор сети] и установить плагин для поиска и замены. Процедура установки аналогична процедуре установки плагина NS Cloner. Один из подходящих плагинов для решения этой задачи — <https://wordpress.org/plugins/search-and-replace>. После установки плагина надо активировать его — тогда его можно использовать на любом подсайте. Итак, выполним некоторые изменения в последнем клоне.

Выбрав панель инструментов для сайта, мы можем получить доступ к инструменту Search and Replace [Поиск и замена] в меню слева. Он расположен в меню Tools > Search and Replace [Инструменты > Поиск и замена]. Выбрав этот пункт меню, перейдите на вкладку Search & Replace. Затем укажите искомое слово и слово, на которое его нужно заменить. После этого выберите таблицы, снимите флажок Dry Run [Тестовый запуск] и выберите Save changes to Database [Сохранить изменения в базе данных]. Наконец, нажмите кнопку Do Search & Replace [Выполнить поиск с заменой].

При выполнении поиска и замены следует соблюдать осторожность, так как это может привести к замене лишних строк. Например, мы хотим заменить слово “dog” на слово “cat”. Однако в этом случае “hotdog” также будет заменено на “hotcat”. После замены слов можно осмотреть результат, и если есть ошибки — исправить их. Другой вариант — сначала выполнить тестовый запуск, чтобы просмотреть, как будут выглядеть фактические изменения, а уж потом внести изменения в базу данных.

На данном этапе мы рассказали обо всех основных действиях, которые необходимо выполнить для обустройства нашего кусочка Интернета с использованием мультисайта. Мы можем настроить сайты, изменить текст и изображения на них и удалить всё, что нам не требуется. Эти навыки очень важны. Чтобы закончить это упражнение, вернемся к исходному сайту и внесем несколько изменений в его внешний вид. Мы можем использовать тему WordPress по умолчанию и изменить ее так, как нам надо.

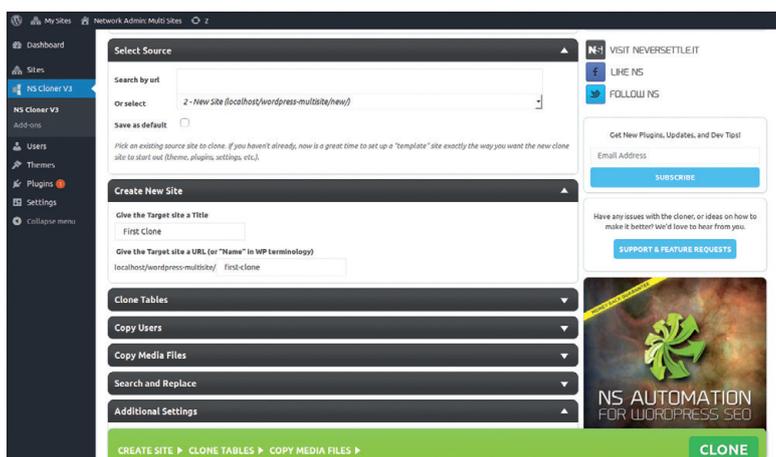
Давайте сосредоточимся на создании мультисайта профессионального вида. Мы можем просто собезьянничать готовый дизайн, не утруждаясь самостоятельным кодированием и правкой рисунков.

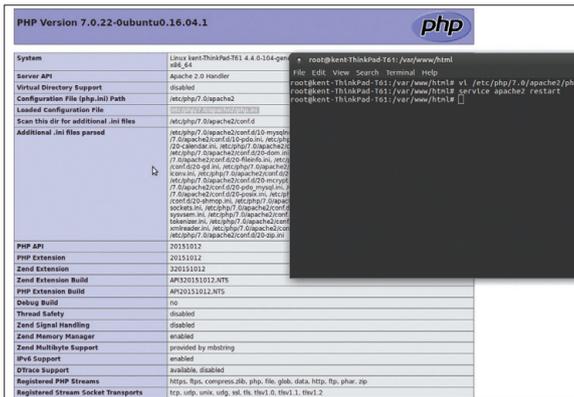
Постепенные изменения

Итак, вернемся к нашему исходному сайту в списке сайтов. Остальные сайты из списка пока можно удалить, потому что мы будем клонировать исходный сайт. Можно удалять сайты по одному или установить флажки рядом с каждым сайтом (кроме первого). Затем щелкните раскрывающийся список Bulk Actions [Групповая обработка], выберите Delete [Удалить] и нажмите кнопку Apply [Применить]. После этого вы перейдете на новую страницу — нажмите кнопку Confirm [Подтвердить].

Теперь давайте слегка изменим внешний вид исходного сайта и откроем панель управления для исходного сайта. Чтобы изменить основной рисунок, откройте раздел Appearance > Themes > Customize [Внешний вид > Темы > Настроить]. Вы увидите список вкладок. Для начала выберите Header Media [Файлы мультимедиа заголовка]. Посмотрев влево, вы увидите, что изображение можно изменить. Чтобы узнать размер текущего изображения заголовка, можно открыть сайт в Firefox, щелкнуть правой кнопкой

В WordPress есть множество инструментов клонирования, включая NS Cloner. Отшлифовав исходный сайт, можно создать копии этого сайта любым удобным способом.





Конфигурацию PHP можно просмотреть с помощью пары-тройки команд. Вы узнаете текущие параметры и расположение файла настройки.

мыши по большому изображению справа и выбрать View Image Info [Показ информации об изображении].

В окне Image Info [Информация об изображении] есть список с несколькими элементами. Нам надо выбрать элемент с именем `http://localhost/wordpress-multisite/wp-content/themes/twentyseventeen/assets/images/header.jpg`. Теперь можно просмотреть его размер и создать новое изображение точно такого же размера. Размер нашего изображения составляет 2000×1200. После создания этого изображения выберите Add New Image [Добавить новое изображение] в разделе Header Media [Файлы мультимедиа заголовка], как было описано ранее. Затем немного обрежьте изображение и нажмите кнопку Publish [Опубликовать]. Если теперь обновить сайт `localhost/wordpress-multisite`, мы увидим в заголовке новое изображение. Новое изображение уже лучше соответствует нашим потребностям, но на сайте всё равно много лишней и ненужной информации: заголовок “Just another WordPress site [Очередной сайт на WordPress]” и всё содержимое боковой колонки, кроме категорий.

Чтобы изменить заголовок, вернитесь в раздел Header Media и перейдите на вкладку Site Identity [Информация о сайте]. Далее просто измените или удалите его. На этой странице также можно изменить название сайта. Потом нажмите кнопку Publish. Чтобы удалить всё ненужное справа, перейдите в главное меню и выберите Appearance > Widgets [Внешний вид > Виджеты]. Здесь можно выбрать любой раскрывающийся список и удалить его. В данном случае категории мы сохраним, потому что у каждого сайта будет свой список категорий. Если мы когда-нибудь захотим вернуть какой-либо список, его можно просто перетащить слева на боковую панель Blog и разместить так, как необходимо.

Неплохо; давайте создадим новую домашнюю страницу. Для этого откройте Appearance > Customize [Внешний вид > Настроить] или Appearance > Themes > Customize [Внешний вид > Темы > Настроить] и выберите Homepage Settings [Параметры домашней страницы]. Затем выберите Static Page [Статическая страница]. В разделе Homepage [Домашняя страница] выберите Sample Page [Пример страницы] и нажмите кнопку Publish [Опубликовать]. Содержимое этой страницы можно изменить на панели управления, выбрав Pages > Edit [Страницы > Изменить] для соответствующей страницы. Затем просто нажмите кнопку Update [Обновить], и наши изменения будут сохранены.

Теперь изменим текст нижнего колонтитула. Мы можем изменить его в шаблоне темы. В нашем случае это файл `var/www/html/wp-content/themes/twentyseventeen/template-parts/footer/site-info.php`. Кроме того, мы можем открыть раздел Appearance

» УСТАНОВКА WORDPRESS

Сначала загрузите WordPress с сайта wordpress.org. Так как WordPress использует базу данных, надо настроить базу данных MySQL с именем пользователя и паролем. Затем распакуйте файл и переименуйте его в `wordpress-multisite`. Получившийся каталог можно загрузить на web-хостинг или на наш сервер в каталоге `/var/www/html` на компьютере Linux.

Далее, откройте в браузере web-страницу, соответствующую нашей папке. В нашем случае это будет `localhost/wordpress-multisite`. На web-хостинге это будет нечто вроде `http://example.com/wordpress-multisite`. Следуйте инструкциям, чтобы указать имя базы данных, имя пользователя и пароль. Префикс по умолчанию — `wp_`, но его можно изменить.

На домашнем сервере потребуется изменить файл `wp-config-sample.php`, указав в нем имя базы данных, имя пользователя и пароль, и сохранить его в файле `wp-config.php`, если сервер не может записать этот файл. В конце установки мы дадим 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.

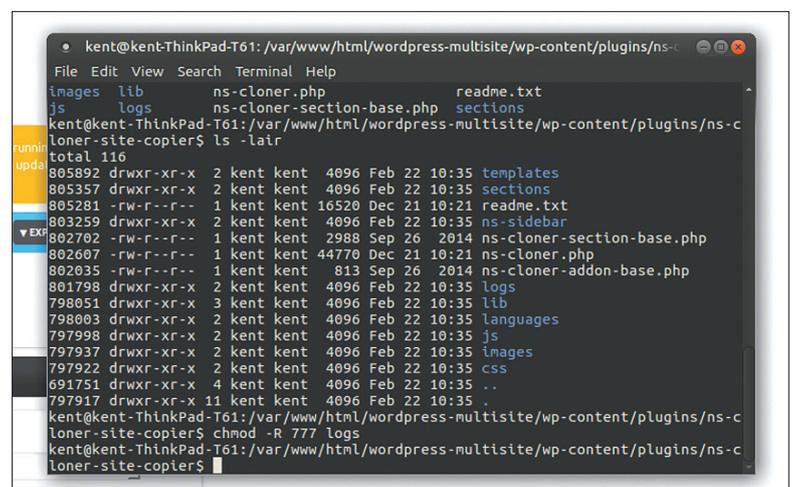
После ввода этих данных и нажатия кнопки Install WordPress [Установить WordPress] открывается окно входа в панель администратора, в котором можно ввести имя пользователя и пароль для входа.

> Customize [Внешний вид > Настроить] и выбрать Additional CSS [Дополнительная CSS]. Следующий код скроет колонтитул из используемой сейчас темы Twentyseventeen:

```
div.site-info{display: none !important}
```

После этого тема почти готова к использованию. Но есть еще две вещи, которые можно изменить. Мы можем создавать и редактировать категории на панели инструментов в разделе Posts > Dashboard [Записи > Панель управления]. Затем можно создать меню — в разделе Appearance > Customize > Menus [Внешний вид > Настроить > Меню]. Управляется меню через раздел Appearance > Menus [Внешний вид > Меню]. Отныне у нас есть хороший исходный сайт, и мы можем клонировать его сколько душе угодно. Вот и займемся.

Теперь мы можем создавать большие мультисайты с единой базой данных. И масштабы нашего творчества будут ограничиваться только нашим воображением. **LXF**



Создать папку и изменить разрешения можно моментально. Если мы хотим записать файлы в папку, необходимо разрешение 777, иначе файл пропадет.

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

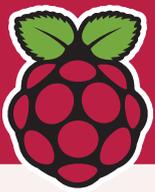
ИНЖЕНЕРНЫЙ ИННОВАЦИОННЫЙ КЛАСС

для обучения информатике, робототехнике и 3D-прототипированию на базе свободного аппаратного и программного обеспечения



Комплектации и цены

НАИМЕНОВАНИЕ	ЦЕНА, руб.	КОМПЛЕКТАЦИЯ, шт.					
		Демонстрационный набор	Инженерный инновационный класс РОББО				
			Экспресс	Стандарт	Делюкс	Расширенный	
РОББО Робоплатформа / СкретчДуино Робоплатформа	23 000	1	6	13	15	15	
РОББО Лаборатория / СкретчДуино Лаборатория	11 500	1	6	13	15	15	
РОББО Схемотехника стандартный набор	4 800	1	6	13	15	15	
Миникомпьютеры Cubieboard	9 500	—	—	—	15	15	
РОББО Цифровая Лаборатория с набором датчиков	90 000	—	—	—	1	1	
РОББО Протос портативный центр прототипирования	400 000	—	—	—	1	1	
3D-принтер РОББО Мини	36 000	1	1	1	1	5	
Стартовый набор расходных материалов для 3D-принтера РОББО Мини (катушка пластика и синий скотч)	2 900	1	1	1	—	—	
Набор аксессуаров и расходных материалов для 3D-печати (6 катушек пластика, синий скотч, инструменты для обслуживания принтера)	14 500	—	—	—	1	1	
РОББО 3D-принтер: конструктор для самостоятельной сборки	30 000	—	—	—	—	6	
СкретчДуино Робоплатформа конструктор для самостоятельной сборки	15 000	—	—	—	—	6	
12 месяцев подписки на Электронный учебно-методический комплекс (ЭУМК)	5–6 лет	40 000	—	—	—	1	1
	7–10 лет	40 000	—	—	—	1	1
	11–15 лет	40 000	0,5	1	1	1	1
	РОББО Лагерь	40 000	—	—	—	—	1
Очное обучение преподавателя на курсе 72 часа	60 000	—	1	1	1	1	
12 месяцев научно-методического сопровождения инновационного проекта в образовательном учреждении	40 000	—	—	—	—	1	
Набор свободного ПО для образования, инженерного творчества и спортивного программирования	бесплатно	1	1	1	1	1	
Онлайн обучение преподавателя	бесплатно	1	1	1	1	1	
СТОИМОСТЬ КОМПЛЕКТА		98 200 руб.	374 700 руб.	649 800 руб.	1 452 500 руб.	1 946 500 руб.	



Пол Бич — соучредитель PiMogoni, компании по разработке и продаже продуктов для производителей, преподавателей и творцов.

» БРОСАЯ КЛИЧ

Странно радуется, когда кто-то говорит «Raspberry Pi? Не слышал об этом». Можно показать им, что бывает, когда люди делают свои проекты с радостью. Редко когда кто-то вдруг не прозревает, увидев позитивный эффект такого способа.

От простых серверов, старомодных игровых автоматов и музыкальных центров до предоставления обучения и науки в места, где это нужнее всего, но ранее никогда такого не имевшие, поскольку это было слишком сложно или дорого.

Но ведь не все учатся, как вы, или любят делать что-то тем же манером. И это нормально, ведь где-то в пушистых облаках сообщества Raspberry Pi 10 разных людей делают что-то десятками разными способами, подстрекаемые желанием рассказать людям о том, что и как можно создать. Энтузиазм, забота и обмен идеями скрывают стальную веру, что таким и должно быть техническое творчество, и нам не нужна оторванная от жизни, высокомерная выдача знаний, при которой мы росли.

Свидетельством служат обычно хороший результат поиска по «Raspberry Pi + что-то еще» (попробуйте «Дверца для кошки» или «Мониторинг растительности») и то, что люди всё еще растут, увлекаются и делятся новым уже шесть лет, от подземных глубин (погуглите «Raspberry Pi SoilCam») до земной орбиты (ищите «AstroPi»).

Может, в следующие 6 лет мы увидим шквал Raspberry Pi, несущий науку по всей планете? Настает время иметь достойную космическую программу, к которой все могут присоединиться.

Что выросло на Малиновых полях

Присоединяйтесь к Raspberry Pi Foundation и другим производителям — праздновать созидание и изучение.

Весь уикенд 30 июня будет проходить фестиваль Raspberry Fields — новое ежегодное событие, организованное Raspberry Pi Foundation на площадке Cambridge



Как-то не очень похоже на поле.

Junction. Для лиц младше 16 лет билеты бесплатные (имейте в виду, что заказывать заранее всё равно придется), а для всех остальных стоимость £5, что, несомненно, оценит сообщество!

Это преподносится как шанс для людей попробовать свои силы в творческой работе с технологиями и отпраздновать всё, что разработчики из Великобритании уже достигли и узнали, благодаря различным мероприятиям типа Code Club, CoderDojo, Raspberry Jam или просто домашнему баловству с компьютером.

Будут шансы узнать больше о многих проектах разработчиков по всей стране, попрактиковаться в деятельности, поучаствовать в научно-технических беседах и насладиться музыкальными выступлениями. Добавьте сюда бесплатную еду, раскраску лиц, веселые представления и многое другое; для каждого возраста найдется что-то свое!

Узнать больше и купить билеты: www.raspberrypi.org/raspberry-fields.

Музей коробок

Учредите мини-музей.

Привлекает к чудесам музеев детей, которые не могут познакомиться с их коллекциями, Museum in a Box [англ. Музей в коробке] использует выполненные на 3D-принтере модели и фигуры в сочетании с NFS-фломастерами и Pi-коробкой, чтобы выставить живые коллекции, которые иначе не увидели бы. Больше о проекте можно узнать здесь: www.museuminbox.org.



М-мм, JAM

Зарядись от праздника умельцев.

- 9 мая** **JAM в Котсуолде** ... здание Waterworth, г. Челтенхем
 - 20 мая** **Jamming в Мальборо** ... Городская ратуша
 - 21 мая** **JAM в Йорке** ... библиотека Acomb Explore
 - 4 июня** **JAM в Престоне** ... Media Innovation Studio, 4-й этаж
 - 6 июня** **JAM в Лидсе** ... Колледж Swallow Hill
- Узнать, какая JAM-тусовка ближе к тебе, можно здесь: www.raspberrypi.org/jam/calendar.



Ищите этот логотипчик!



ExaGear Desktop

В своей неизменной погоне за расширением возможностей Pi, Лес Паундер отдал бы ARM, а то и целую армию, лишь бы запустить код x86 на Raspberry Pi.

ВКРАТЦЕ

Эмулятор виртуальной машины, предлагающий возможность запуска 32-разрядного кода x86 на процессорах ARM, например, на Pi. Он работает с родными приложениями Linux, а также с WINE для совместимости с Windows, и предоставляет базовый инструмент, который удовлетворяет потребностям тех, кто должен использовать определенную программу на Raspberry Pi.

Процессор x86 имеется в наших ноутбуках, настольных компьютерах и в наших центрах обработки данных. Но для мобильных устройств и Raspberry Pi на рынке доминируют процессоры ARM, благодаря их низкому потреблению энергии и соответствующей вычислительной мощности. Приложения скомпилированы под архитектуру (ARM/x86) того устройства, на котором они предназначены для использования; ну, а если вам надо запускать на процессоре ARM приложение x86? ExaGear Desktop от Eltechs — эмулятор виртуальной машины, который, как утверждается, запускает «...любой вид программного обеспечения практически на любом оборудовании», и он запускает программное обеспечение x86 на процессоре ARM, хотя и с определенными ограничениями.

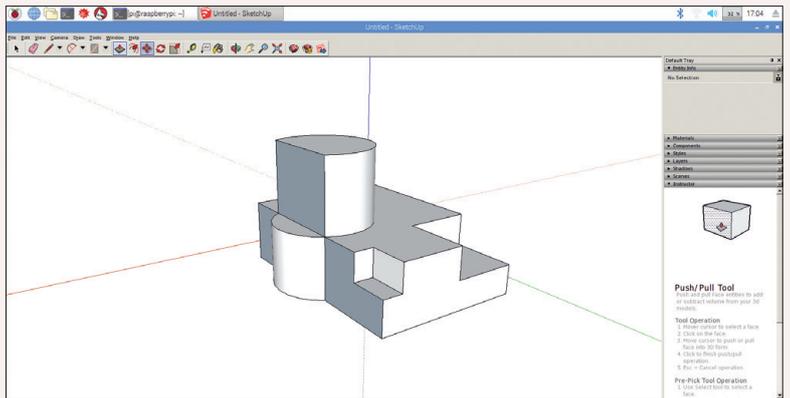
Установка безболезненна, но требует оплаты около £21 за лицензию на один экземпляр Raspberry Pi 3. Как только ваш платеж пройдет, вы получите по электронной почте ссылку для загрузки своей копии программного обеспечения и файла лицензионного ключа.

Запуск и работа

Для запуска ExaGear Desktop мы просто нажимаем значок на рабочем столе Raspbian. Открывается специальный терминал, используемый для выполнения 32-разрядных x86-приложений Linux. Здесь мы можем запускать и устанавливать приложения x86 и родные приложения Linux. Например, Sublime Text 3 работает очень хорошо, хотя его загрузка занимает изрядное время даже с самым мощным Pi. Мы попробовали последнюю версию Firefox Quantum (32-разрядная версия x86), и хотя он показался многообещающим, его не удалось запустить из-за проблемы с libgtk3, которую не удалось устранить даже после установки.

ExaGear Desktop также работает со слоем совместимости WINE, позволяющим запускать приложения Windows под Linux. Имея это в виду, мы установили библиотеки WINE, а затем загрузили несколько приложений с www.portableapps.com.

Популярный IDE/текстовый редактор для Windows Notepad++ работает довольно хорошо, только немного подвисает. Zint Barcode Studio работала очень хорошо, и за несколько минут мы насоздавали штрих-коды. Затем мы протестировали Google Sketchup 2016, довольно громоздкое приложение, которое съело более половины оперативной памяти на Pi и отняло большую часть циклов процессора. Sketchup работал, хотя отнюдь не быстро,

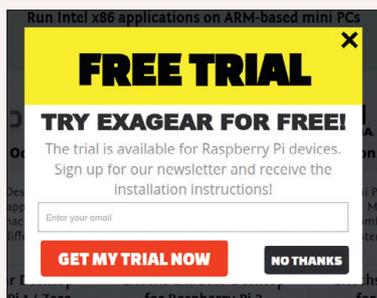


Благодаря ExaGear Desktop стал возможным запуск приложений x86 (Linux и Windows) на Pi.

часто замирая на несколько секунд между действиями. Но, во всяком случае, работал.

Мы попытались запустить несколько игр начала 2000-х с ExaGear Desktop и типичной настройкой WINE, но все попытки были сорваны, так что сегодня классической игровой сессии начала XXI века не произошло. Мы смогли сыграть в Alien Trilog (1996 г.), но игра больше напоминала слайд-шоу, чем клон Doom.

Так зачем заморачиваться с ExaGear Desktop? Ну, если у вас стынет устаревшее 32-разрядное приложение x86 для Linux, не портируемое на ARM, то это удобное решение. Но ограничение ExaGear таково, что он подходит для небольших приложений, особенно в случае с Windows. Конечно, вы можете загрузить Photoshop 2005 г., но с ним будет мало удовольствия. На форумах ExaGear Desktop предлагается поддержка и решения общих проблем, и с их помощью вы можете настроить его для удовлетворения ваших потребностей. Воспользуйтесь бесплатной пробной версией и проверьте приложения, которые собираетесь использовать, прежде чем расстаться со своими деньгами. LXF



ExaGear Desktop предлагает бесплатную пробную версию программного обеспечения, и ее надо попробовать, прежде чем расстаться со своими деньгами. Убедитесь, что ваше приложение поддерживается и работает как полагается!

ВЕРДИКТ

РАЗРАБОТЧИК: Eltechs
САЙТ: www.eltechs.com
ЦЕНА: £ 11 Pi 1/Zero, £ 16 Pi 2, £ 22 Pi 3

ФУНКЦИОНАЛЬНОСТЬ	6/10
ПРОИЗВОДИТЕЛЬНОСТЬ	6/10
УДОБСТВО В РАБОТЕ	6/10
ОПРАВДАНОСТЬ ЦЕНЫ	7/10

Это не идеальное решение, но если вам позарез нужна какая-то программа для работы на Pi, ExaGear Desktop ее предоставит.

»» **Рейтинг 6/10**

Набор молодого бойца

Будь прокляты очумелые ручки и морока с электроникой: **Алекс Кокс** тестирует готовый к бою набор от Maker Life для программирования Pi обычным смертным.

ВКРАТЦЕ

Полный набор с Pi для начинающих, включая Pi W v1.1 с припаянными контактами, набор светодиодов, резисторы, две кнопки, маленькую макетную плату, предустановленную карту microSD на 16 ГБ, USB-концентратор и кабель HDMI с адаптером.

Raspberry Pi — идеальная платформа для экспериментов с оборудованием. Это в некотором роде ясно. Но сколько из этих вездесущих мини-плат работает с абсолютно не измененной контактной частью? Мы бы побились об заклад, что подавляющее большинство, и, как показывает лучший базовый комплект Maker Life, вообще-то не так сложно управлять выводами и считать входы с точки зрения как с аппаратной, так и программной.

Getting You Started Kit поставляется буквально со всем необходимым для начала работы с электроникой. Там есть Raspberry Pi Zero W, карта MicroSD с загруженным *Noobs*, все кабели и адаптеры, необходимые для питания и подключения к монитору и периферийным устройствам, а также небольшой комплект компонентов.

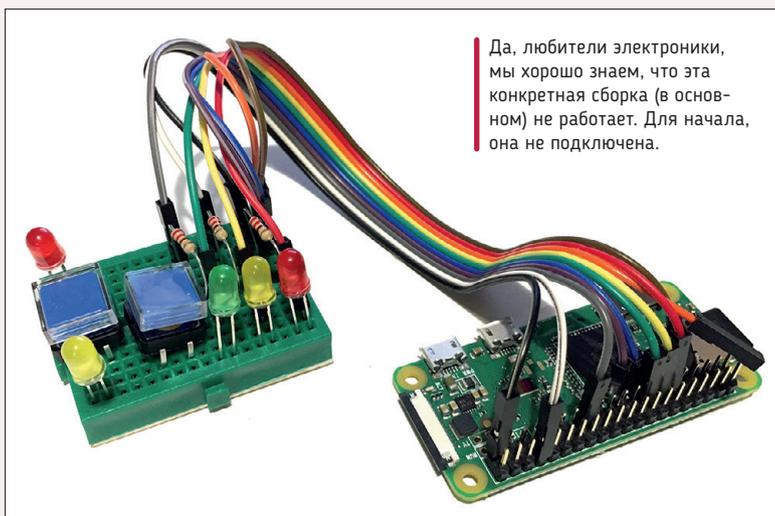
«Небольшой» здесь ключевое слово. Это не набор для создания грандиозных проектов. Он больше упирает на возможности и на демонстрацию простоты требуемого кода. По части экспериментального оборудования, вам предоставляется небольшая макетная плата, пригоршня резисторов по 220 Ом, несколько светодиодов и несколько кнопок, а также предустановленный на Pi Zero W штыревой разъем и 10-жильный плоский кабель мама—папа. Этот комплект избавляет от многих мучений первых шагов — в том числе буквальных, если вы не умеете держать пальник; а значит, вы можете резвиться с ним, как вам угодно.

Чтение по ходу

Жизнь и смерть любого комплекта, подобного этому, зависит от его руководства пользователя, и в Maker Life проделали отличную работу по представлению трех проектов — версии светофора, управляемый кнопкой мигающий светодиод и регулируемый пешеходами перекресток — причем очень разумным образом. Поскольку руководство проводит разбор кода, (всё на Python), то ясно, что вы делаете и почему.

Хотя проекты достаточно легко пройти, просто набрав код (мы готовы поклясться, что взрослый при известном умении пройдет весь набор за час), части кода по возможности повторяются, хорошо сформулированы и превращают этот комплект в полностью доступный для детей. Речь идет не только о синтаксисе и библиотеках: набор Getting You Started Kit действительно опирается на самые основы, представляя по ходу дела набор главных концепций кодирования, а употребление Python означает, что их легко понять в принципе.

Тем не менее, за пределами пошаговой части оно не совсем идеально. Фактические соединения электронных компонентов не всегда четко объясняются, заставляя полагаться на мультяшные иллюстрации и единственную жаргонную схему контактов GPIO Pi, чтобы всё было выстроено в линию. Некоторое дополнительное объяснение того, что на самом деле представляют



Да, любители электроники, мы хорошо знаем, что эта конкретная сборка (в основном) не работает. Для начала, она не подключена.

отдельные контакты, могло бы добавить комплекту немного долговечности, равно как и некоторые предложения путей дальнейшего применения самих проектов. А в его теперешнем виде, стоит вам пройти третий проект, как руководство внезапно заканчивается, оставляя вас на волю волн.

Но давайте отдадим ему должное. Нет ничего более вдохновляющего, чем увидеть, как этот маленький светодиод загорается в первый раз, зная, что это сделали вы. Это первый шаг на долгом пути. И будь руководство утомительно-подробным насчет следующих шагов, возможно, оно не вдохновило бы детей читать и учиться дальше или играть с этими переменными и смотреть, что происходит.

Набор достаточно привлекателен, чтобы вы захотели большего, а не просто пожали плечами, и это прекрасно. Набор от Maker Life — это легкий заботливый вход в мир управления оборудованием. **LXF**

ВЕРДИКТ

РАЗРАБОТЧИК: Maker Life
САЙТ: www.makerlife.co.uk
ЦЕНА: £50

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

Отличный набор для начинающих, как детей, так и взрослых, с простыми (по большей части) инструкциями и лаконичным набором компонентов.

» **Рейтинг 8/10**

ARDUINO

Объедините Arduino с Raspberry Pi

Лес Паундер показывает, что Raspberry Pi и Arduino могут крепко подружиться и работать вместе в проекте, который включает последовательные данные.



НАШ ЭКСПЕРТ

Лес Паундер — независимый разработчик, сотрудничающий с такими организациями, как Raspberry Pi Foundation, чтобы содействовать развитию навыков умельцев. Вы найдете его блог на www.bigl.es.

ВАМ НУЖНЫ

- > Arduino Uno или совместимый
- > Любая модель Raspberry Pi с последней Raspbian OS
- > Макетная плата датчика TMP36
- > Три проволочных перемычки папа-папа
- > Весь код и схему можно скачать с <http://bit.ly/LXF237-serial-project>

A rduino и Raspberry Pi — фантастические машины, и здесь мы воспользуемся Arduino для считывания показаний температурного датчика TMP36. Мы отправим данные через последовательное USB-соединение на Raspberry Pi, где запущен код Python, который будет читать и интерпретировать данные, а затем действовать на их основе. Этот проект покажет, что две машины могут сосуществовать и что Arduino пригоден для расширения возможностей Pi.

Подготовка Arduino

Чтобы установить *Arduino IDE*, перейдите на www.arduino.cc/en/Main/Software и выберите версию для Linux ARM. Загрузите программное обеспечение и извлеките архив в домашний каталог (`/home/pi/`). Затем откройте окно терминала. Теперь в вашем домашнем каталоге должен быть каталог **arduino-1.8.5**. В терминале измените каталог, чтобы мы оказались в каталоге Arduino:

```
$ cd arduino-1.8.5
```

В каталоге должен быть файл с именем **install.sh**, который установит *Arduino IDE* на Pi. Для запуска файла введите

```
$ sudo ./install.sh
```

Через несколько секунд установка завершится. Следующим шагом будет выдача Pi доступа к устройству Arduino и отправка ему кода, посредством добавления пользователя Pi в группу *dialout* с помощью команды *usermod*:

```
$ sudo usermod -a -G dialout pi
```

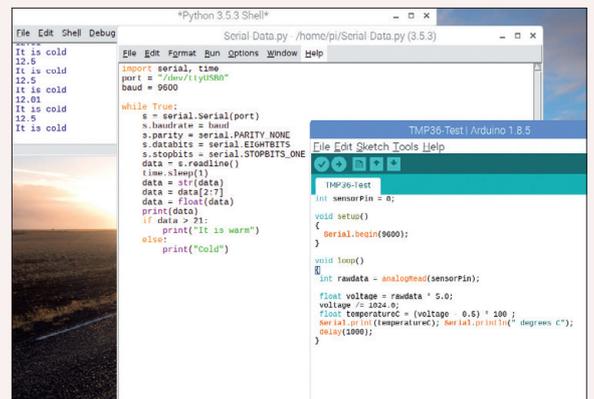
Наконец, перезагрузите Raspberry Pi.

Сборка оборудования

Датчик TMP36 имеет только три вывода, и, если смотреть со стороны выводов (плоская сторона), то порядок такой: Vcc, Vout и GND [Земля]. TMP36 может работать с напряжением от 2,7 до 5,5 В, поэтому напряжение 5 В с Arduino обеспечит питание TMP36, а контакт GND подключается к GND на Arduino. Наконец, вывод Vout (в центре) соединяется с A0 на Arduino. Для получения более подробной информации загрузите схему для этого проекта. Проволочными перемычками папа-папа обеспечьте эти соединения, а затем подключите Arduino к USB-порту Raspberry Pi.

Пишем код Arduino

Чтобы открыть *Arduino IDE*, перейдите в главное меню, выберите Programming > Arduino, и через несколько секунд приложение загрузится и представит шаблон для нашего кода. Удалите шаблон (нажмите Ctrl+A, а затем клавишу Delete), чтобы начать код



Этот код можно использовать для чтения последовательных данных с любого устройства, подключенного к Pi.

зано. На Arduino мы пишем код в приложении, компилируем его и затем записываем код в Arduino. Это называется заливкой.

Наша первая строка кода создает переменную. Это объект хранения данных, позволяющий нам ссылаться на него по имени и отображать содержимое объекта. В данном случае мы создаем переменную с именем **sensorPin**, которая на самом деле является соединением Vout TMP36 с контактом A0 Arduino. Мы должны указать Arduino, какой тип данных будет содержать переменная. В нашем случае это целочисленное значение 0.

```
int sensorPin = 0;
```

Затем мы создаем раздел кода, настраивающего Arduino на задачу. Этот код запускается один раз в начале. Здесь мы сообщаем Arduino, что хотим использовать последовательное соединение со скоростью 9600 бод.

```
void setup()
{
  Serial.begin(9600);
}
```

Чтобы Arduino непрерывно выполнял следующий код, надо поместить код внутри цикла. На следующем шаге мы создаем еще одну переменную, которая будет считывать необработанные показания с аналогового вывода A, а затем сохранять их как целое число внутри переменной *rawdata*:

```
void loop()
{
  int rawdata = analogRead(sensorPin);
```

Мы всё еще находимся внутри цикла, и теперь надо применить немного математики для преобразования необработанных данных в новую переменную с именем `voltage`, которая будет содержать значение с плавающей точкой. `Voltage` — это наши необработанные данные, умноженные на 5,0 (напряжение, которое мы подаем на TMP36; измените это на 3,3 В при использовании 3,3 В Arduino). Затем мы употребляем составной оператор для деления значения, сохраненного в переменной `voltage`, на 1024 для 10-битного аналога, используемого в Arduino.

```
float voltage = rawdata * 5.0;
```

```
voltage /= 1024.0;
```

Новая переменная, с именем `temperatureC`, используется для хранения решения уравнения, в котором сначала вычитается 0,5 из переменной `voltage`, а результат умножается на 100, что дает нам температуру в градусах Цельсия:

```
float temperatureC = (voltage - 0.5) * 100 ;
```

В последнем разделе кода мы выводим данные температуры на последовательную консоль, а также выводим единицу измерения. Задержка на секунду помогает сделать код читаемым:

```
Serial.print(temperatureC); Serial.println("degrees C");
```

```
delay(1000);
```

```
}
```

Сохраните код! Затем перейдите в `Tools > Board` и убедитесь, что ваш Arduino соответствует в данном случае `Arduino/Genuino Uno`. Затем перейдите в `Tools > Port` и убедитесь, что выбран `ttUSB0`. Теперь нажмите `Sketch > Upload` для заливки на Arduino. Это займет около минуты. После этого мы можем перейти к коду Python.

Код Python

Откройте редактор Python 3, который находится в меню `Programming`. Нажмите `File > New`, а затем сохраните новый файл как **Serial-Data.py**. Сохраняйте почаще, чтобы избежать потери данных.

Мы начинаем код с импорта двух библиотек кода. Это `serial`, используемая для последовательного соединения, и `time`, используемая для регулирования скорости проекта.

```
import serial, time
```

Затем мы создаем две переменные — `port` и `baud`, которые используются для определения последовательного порта и скорости соединения:

```
port = "/dev/ttyUSB0"
```

```
baud = 9600
```

Наш основной цикл будет постоянно проверять последовательный порт на данные с указанной нами скоростью:

```
while True:
```

```
    s = serial.Serial(port)
```

```
    s.baudrate = baud
```

Гарантируем правильность отправленных данных и отсутствие ошибок, проверяя их четность:

```
s.parity = serial.PARITY_NONE
```

```
s.databits = serial.EIGHTBITS
```

```
s.stopbits = serial.STOPBITS_ONE
```

Потом прочитаем последовательные данные и на секунду остановимся:

```
data = s.readline()
```

```
time.sleep(1)
```

Следующие четыре строки преобразуют полученные данные в строку и вырезают из строки нужные нам сведения (отсчитывая с нуля, мы находим начало и конец данных температуры); затем `data` преобразуется обратно во `float` и данные выводятся в оболочку.

```
data = str(data)
```

```
data = data[2:7]
```

```
data = float(data)
```



Эти платы обычно имеют последовательный интерфейс, поэтому все они могут быть использованы в проекте.

```
print(data)
```

В последнем разделе мы создаем тест, который сравнивает значение, хранящееся внутри переменной `data`, с жестко закодированным значением, в данном случае — 21°C. Если температура выше 21°C, то выводится «Тепло», а если меньше, то выводится «Холодно».

```
if data > 21:
```

```
    print("Тепло")
```

```
else:
```

```
    print("Холодно")
```

Сохраните код и нажмите `Run > Run Module (F5)`, чтобы его запустить. Теперь мы должны увидеть данные температуры, появляющиеся в оболочке Python, и соответствующее сообщение, зависящее от того, насколько тепло или холодно.

Ну вот, мы научились подключать Arduino к Raspberry Pi и извлекать данные из последовательного порта с помощью Python. Отличная работа! **LXF**

СКОРАЯ ПОМОЩЬ

Если вы не видите свой Arduino как подключенное устройство, выясните командой `dmesg`, что происходит за кулисами на Pi. Команда покажет, где подключены устройства, и все ошибки, которые могут иметь место.

» ПОСЛЕДОВАТЕЛЬНЫЙ ИНТЕРФЕЙС – МАСТЕРАМ

Последовательный интерфейс является одним из самых основных видов связи, поэтому неудивительно, что многие устройства также имеют последовательные порты. BBC micro:bit имеет последовательный интерфейс, а для использования интерфейса с MicroPython на micro:bit всё, что нужно сделать, это ввести данные в оболочку Python, которая затем выведет их последовательно — не нужно никаких заковыристых команд:

```
from microbit import *
while True:
    print("LXF ROOLZ OK")
    sleep(1000)
```

Теперь нужно только отредактировать имя порта и скорость соединения:

```
port = "/dev/ttyACM0"
baud 115200
```

Внутри цикла нам еще надо определить последовательный порт и скорость передачи, затем проверить четность и другие параметры. Затем мы считываем последовательные данные в переменную `data` и выводим переменную в оболочку, подтверждая, что наш micro:bit связывается с подключенным Pi через последовательный порт. Для иллюстрации мы поместим пример такого кода в загрузку для нашего проекта. Мы рассматривали это более подробно в **LXF209**, где использовали micro:bit в качестве контроллера жестов для *Minecraft*.

НАТАРИ

Эмуляция Atari ST на Raspberry Pi

Ник Пирс нацепил мешковатые джинсы и тельняшку и отплыл назад во времени — показать, как превратить ваш Pi в полноценную машину Atari.



НАШ ЭКСПЕРТ

Ник Пирс видит, что данной статьей замыкается полный круг его карьеры. Свою писательскую карьеру он начал в качестве редактора диска в журнале *ST Format* еще в 1994 г. Это было бы похоже на возвращение домой, если бы он мог вспомнить столь давние времена.

Вы почти наверняка помните конец 1980-х, когда восьмибитная революция в вычислительной технике постепенно сменилась взлетом 16-разрядных машин. Atari и Commodore были на переднем крае этой революции, а в конечном счете победила Amiga. Но некоторое время в конце 1980-х и начале 1990-х гг. Atari ST была к ней близка. И благодаря блестящему *Hatari* (<http://hatari.tuxfamily.org>), вы можете теперь воскресить ST (и его преемников) на Raspberry Pi.

Первые шаги

Для запуска *Hatari* вам понадобится среда рабочего стола — мы используем Raspbian PIXEL. Последняя версия *Hatari* использует SDL2 для рендеринга графики, так что вам надо включить драйвер OpenGL, учтя, что он всё еще экспериментальный и бывает немного нестабильным. Это также означает, что для оптимальной работы *Hatari* требует Pi 2 или выше.

Чтобы включить драйвер, откройте окно терминала, введите `sudo raspi-config` и нажмите Enter. Выберите `Advanced Options > GL Driver > GL (Full KMS)` — рабочий стол OpenGL с полным KMS. По запросу перезагрузитесь.

Далее установка проста: откройте `Pi > Preferences > Add/Remove Software`, введите 'hatari' в поле Search, и вы увидите одно совпадение: отметьте его и нажмите OK, чтобы установить. После установки откройте меню `Pi > Games` — вы увидите две записи: *Hatari* и *Hatari UI*. Выберите первый вариант, где вы сразу же столкнетесь с сообщением о невозможности загрузить файл TOS.

TOS является операционной системой Atari и поставляется с различными ревизиями и версиями, в зависимости от того,

какую 16-разрядную машину Atari вы пытаетесь эмулировать. Технически вы должны поставить TOS сами, но, к счастью, трудная работа по получению любой версии TOS была сделана за вас.

Прежде чем продолжить, сделаем краткое замечание о законности: при запуске программы эмулятора в большинстве случаев предполагается, что вы владеете компьютером-оригиналом, а также всем программным обеспечением, запускаемом на нем. С юридической точки зрения вы должны уважать закон, но на практике ничто, кроме вашей совести, не помешает вам поступать, как нравится, даже если машина больше не зарыта у вас на чердаке.

Начнем с получения TOS для *Hatari*. Откройте web-браузер и перейдите на www.atariworld.org/tos-rom, где найдется любая версия TOS, от исходной 1.0 до TOS 4.92 для Atari Falcon. Которая вам нужна? Это зависит от того, что вы собираетесь эмулировать. Исходная модель STFM, которая охватывает большинство баз, лучше всего работает на TOS 1.4, но для некоторых старых игр можно загрузить TOS 1.02. Для эмуляции STE нужна TOS 1.62, а TOS 2.06 (как STFM, так и STE) — хороший выбор при копировании большого количества файлов через рабочий стол ST GEM.

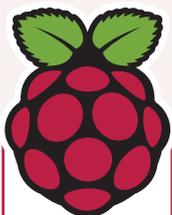
Мудрое слово: файлу TOS 2.06 ROM назначены неправильные права. Надо щелкнуть правой кнопкой мыши по файлу **IMG** после его извлечения и выбрать `Properties > Permissions`. Убедитесь, что `View Content` установлено в `Anyone`, а `Change Content` — в `Only owner`, прежде чем нажать OK и скопировать файл.

Если вы решили остаться на стороне закона и у вас нет Atari, собирающего пыль на чердаке, попробуйте вместо этого EmuTOS (<http://emutos.sourceforge.net/en>). Это бесплатная версия TOS, способная работать на физических компьютерах с дискеты, как и ваш новый эмулятор *Hatari*; но учтите, что игры и программы, которые вы сможете запустить на нем, определяются методом проба и ошибок.

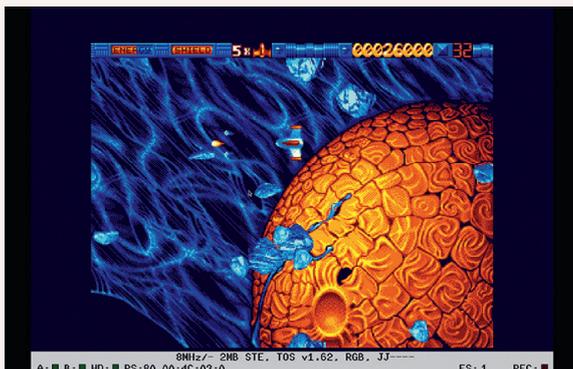
Доходя до загрузки определенной версии, выберите ссылку English UK. ZIP-файл будет сохранен в папке **Downloads**, готовый к использованию. Теперь у вас есть все элементы, необходимые для настройки самого *Hatari*. Пошаговое руководство (стр. 97) показывает основные шаги, которые необходимо выполнить.

Скачайте образы дискеты

Итак, ваша установка *Hatari* теперь стала полнофункциональной машиной Atari — не хватает только программного обеспечения. Большинство игр поставляются на дискетах, и опять-таки кропотливый процесс конвертирования дискет ST в файлы образов, распознаваемых Pi, был проделан другими. Всё, что вам надо



СОВМЕСТИМОСТЬ
ТОЛЬКО
2/3



Atari STE была способна на лучшую графику, как ясно показывает эта игра — *Stardust*.

сделать — это найти и загрузить требуемый образ, они поставляются в формате MSA, ST или STX.

Имеется широкий спектр сайтов, предлагающих образы — погуглите “Atari ST ROM” для получения длинного списка. Если вы справитесь с французским (*Chromium* предложит вам перевести страницы), нам нравится, например, www.planetemu.net/roms/atari-st-games-st. Здесь вы найдете несколько игр, перечисленных несколько раз в зависимости от того, кто предоставил образы — попробуйте один, затем скачайте другой, если не сработает.

Как и в случае с ROM [ПЗУ], образы гибких дисков загружаются в формате ZIP, поэтому сначала вам нужно извлечь их, и вам будут представлены файлы в формате .ST или .STX (предусмотрительно связанные с *Hatari*, так что вы увидите симпатичный значок дискеты ST, помогающий идентифицировать их). Их можно сохранить в любом месте на вашем жестком диске, но мы рекомендуем создать специальную папку — скажем, *atarist* — внутри вашей домашней папки для удобства доступа.

Играйте на ST

Существует два способа загрузки дискет в *Hatari*. Самый простой — найти папку, содержащую ваши диски, и дважды щелкнуть файл .ST или .STX. *Hatari* запустится, используя конфигурацию, сохраненную вами в *hatari.cfg*, а затем — после паузы, которая может быть довольно длинной — появится знакомый зеленый рабочий стол ST или автоматически загрузится игра. При необходимости дважды щелкните по значку диска A, а затем по файлу игры, чтобы начать ее загрузку.

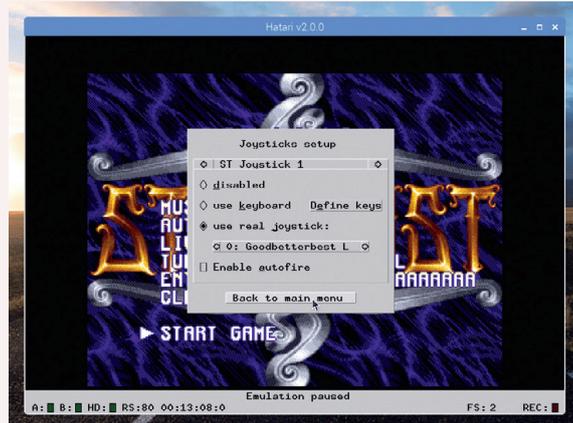
Если вы планируете играть в игру, поставляемую на двух дискетах, проще всего начать прямо внутри самого *Hatari*: нажмите F12, чтобы открыть экран конфигурации, и щелкните по Floppy disks. Нажмите Browse рядом с Drive A, чтобы загрузить диск 1 на диск A, а затем повторите с Drive B для диска 2. Оставьте отмеченным Auto insert B и нажмите Return для возврата в главное меню, а затем OK.

Следите за строкой состояния внизу экрана, которая, мы надеемся, отображает, что дискета по-прежнему загружается. Этот процесс может быть медленным, но просмотрите «Скорую помощь» (стр. 96) на предмет попробовать некоторые возможные настройки. После этого должен появиться главный экран игры, и тогда будет возможность навигации с помощью мыши, клавиатуры или игрового контроллера — если у вас есть подключенный и настроенный.

Чтобы настроить последний, нажмите F12 и выберите Joysticks. Здесь вы можете выбрать, нужно ли отключать порт, использовать «настоящий» джойстик или определить клавиши, используемые для имитации джойстика, а именно вверх, вниз, влево, вправо и пальба. Можно настроить до четырех портов — два обычных джойстика ST плюс два дополнительных джойстика только для STE.



В любое время можно нажать F12, чтобы приостановить *Hatari* и получить доступ ко всей полноте параметров его настройки.



При использовании реального джойстика любой игровой контроллер, подключенный и распознанный Pi, должен быть сразу виден — например, наш проводной контроллер PC/PS3 Giteck обнаружился мгновенно. Как вы видели при определении клавиш для имитации джойстика, джойстики ST были основными и поддерживали только четыре направления движения и одну кнопку пальбы — используйте аналоговый джойстик и верхнюю кнопку пальбы для эмуляции джойстика, а дополнительная кнопка пальбы эмулирует клавишу пробела.

Создайте свои чистые дискеты

Если вы пробуете программы, появившиеся на дисках для журналов — да, мы говорим о *ST Format* — тогда, скорее всего, они были сжаты, чтобы разместить больше контента на диске. Вам нужно будет извлечь их на отдельный «диск» — мы рассмотрим жесткие диски через мгновение, но если вы просто хотите запустить их из виртуального образа дискеты, нажмите Create blank image на экране конфигурации Floppy Disks. Оставьте настройки

К *Hatari* можно подключить реальный игровой контроллер или использовать клавиатурную эмуляцию для игр, требующих джойстика.

СКОРАЯ ПОМОЩЬ

По умолчанию машина Atari эмулирует цветной RGB-монитор, позволяющий переключаться между низким и средним разрешениями. Чтобы запускать программы с высоким разрешением, такие как DTP, нажмите F12 и щелкните экран Atari, затем выберите Mono.

» РЕШЕНИЕ ПРОБЛЕМ СОВМЕСТИМОСТИ

Всё это выглядит просто, и если вам повезет, вы сможете сыграть в несколько игр, прежде чем наткнуться на ту, которая, похоже, не работает. Начните с посещения <http://bit.ly/hatari-compat> для неполного списка программ ST, про которые известно, что они вызывают проблемы. В противном случае проверьте сайт, с которого вы загрузили диск, чтобы узнать, есть ли какая-либо конкретная конфигурация, которую следует использовать. Если вы запускаете более старую игру, попробуйте переключиться в режим ST с TOS 1.02 для обеспечения совместимости.

Не выделяйте под виртуальный ST более 2 МБ ОЗУ, поскольку синхронизация также может вызвать проблемы. 1 МБ для большинства игр должно быть достаточно. В некоторых играх может потребоваться NTSC (US English), а не PAL (UK English) TOS по умолчанию, который вы загрузили, — просто замените их и посмотрите, повезет ли вам теперь.

Некоторые игры и демки очень придирчивы насчет эмуляции дискет — нажмите F12, щелкните по Floppy Disks и снимите флажок Fast floppy access, чтобы узнать, могут ли они теперь успешно загружаться. Другие параметры настройки включают снятие флажков Boot Faster и Patch Timer-D в System. Если налицо искажение экрана, попробуйте также различные настройки синхронизации видео.

Переключение настроек вручную может быть затруднительным, но *Hatari* позволяет сохранять несколько файлов конфигурации, изменяя имя файла .cfg и устанавливая несколько настроек (например, STFM, STE и Falcon), и переключаться между ними. По умолчанию *Hatari* всегда будет при запуске загружать файл *hatari.cfg*, но вы можете быстро переключиться, нажав F12, а затем Load config, чтобы выбрать другой.

СКОРАЯ ПОМОЩЬ

Один из способов уменьшить время загрузки – это отметить Fast floppy access в Floppy disks в конфигурации *Hatari*. Другим является переключение в режим Fast Forward – нажмите AltGr+X, чтобы немного ускорить работу. Для возврата к обычной скорости нажмите это еще раз.

по умолчанию — 80 треков, девять секторов и две стороны — как есть, задайте вашей дискете метку и нажмите Create.

Затем выберите, где сохранить дискету — лучше всего в вашей папке по умолчанию — и переименуйте ее для игры, которую вы планируете скопировать на нее. Нажмите OK, затем Back, затем Drive B:, чтобы вставить ее в диск B. Теперь вы можете скопировать сжатый файл, прежде чем дважды щелкнуть по нему, чтобы распаковать игру или использовать встроенную программу для извлечения файлов на новый диск. Будьте осторожны, это медленный процесс, и вы можете погрузиться в воспоминания... или, возможно, это во мне говорит бывший редактор диска. Кстати говоря, зайдите на www.exxoshost.co.uk/atari/ST/, чтобы получить полный список и скачать ссылки на диски *STF*.

Дополнительно

Теперь у вас теперь есть полностью работающая машина Atari, работающая на приличной скорости. Если вы хотите только вновь пережить былую славу, у вас должно быть всё, что нужно для игры на вашем Atari. Но вы также можете расширить горизонты.

Во-первых, не рассматривайте ST как простой игровой автомат — он также известен как музыкальный инструмент, а когда

рынок игр падает, основные приложения продолжают развиваться — от редактирования видео и обработки текстов до настольных издательских систем. Посетите www.planetemu.net/machine/atari-st, где у вас будет доступ к более широкому спектру наименований, включая демо, диски к журналам, общедоступные выпуски и приложения.

По мере роста количества наименований вы можете устать от необходимости нажимать F12, чтобы менять дискеты. Используйте AltGr+D для смены дисков, а затем AltGr+R, для выполнения горячей перезагрузки, или AltGr+C — для холодной перезагрузки. Кроме того, почему бы не подключить виртуальный жесткий диск к ST — во врезке показан простейший способ добавить один 80-МБ жесткий диск ACSI. Вы можете дублировать этот пустой файл образа для создания нескольких жестких дисков, но подключать можно только по одному.

Другим вариантом является GEMDOS, которая позволяет монтировать папку на вашем хранилище Pi в качестве виртуального диска, упростив копирование файлов прямо на диск из хранилища Pi. Учтите, однако, что вам понадобится либо специальный драйвер жесткого диска, либо доступ к диску с помощью EmuTOS, поддерживающей GEMDOS изначально.

Если вы попадаете в старую игру или две, вы можете захотеть узнать, как сохранить свои достижения. Если в самой игре есть опция сохранения, вы всегда можете создать образ пустой дискеты, как описано ранее, и вставить его в дисковод B. В порядке альтернативы, можно использовать функцию снимков памяти *Hatari*, которая, очевидно, быстрее.

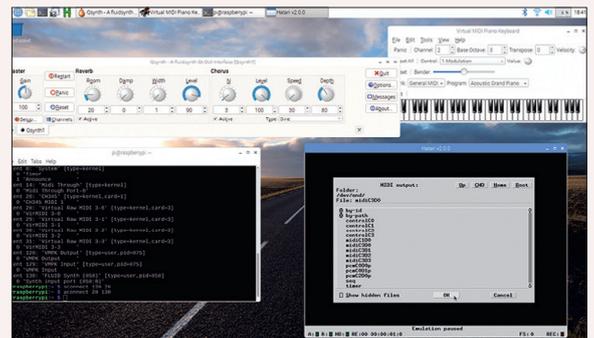
Начните с нажатия F12 и выбора Memory. Нажмите Save, введите подходящее описательное имя файла (с крайне важным расширением **.sav**) и нажмите OK. Вернитесь в игру и в любое время нажимайте AltGr+K, чтобы обновить сохраненный файл с вашим последним достижением, или AltGr+I, чтобы загрузить последнее сохраненное состояние.

Вы также можете в любой момент сделать скриншоты — нажмите AltGr+G, чтобы захватить экран в формате PNG. Нажмите AltGr+A для записи видео в формате AVI (вы увидите, что в строке состояния *Hatari* загорится красным цветом кнопка Rec), и нажмите ее еще раз, чтобы остановить запись. Наконец, нажмите AltGr+Y, чтобы начать и остановить запись звука ST. Все записанные материалы найдутся в домашней папке.

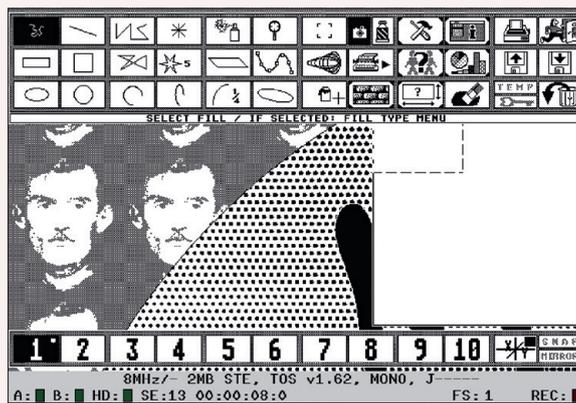
MIDI для музыкантов

ST был очень привлекателен для музыкантов своими встроенными портами MIDI и использовался крупными артистами, включая таких, как Queen — для *The Miracle*. Если у вас имеется кабель USB-MIDI, то можно подключить MIDI-клавиатуру для использования с чем-нибудь вроде *Cubase*.

Во-первых, вам надо проверить, что ваш кабель обнаружен в Raspbian. Откройте окно терминала и введите следующее:



Вооружившись знаниями, вы можете настроить эмуляцию MIDI на своем Atari ST — с помощью либо соответствующего оборудования, либо виртуального ПО.



Ваш виртуальный Atari ST может также эмулировать старый одноцветный дисплей высокого разрешения – два монитора по цене одного!

» СОЗДАНИЕ ВИРТУАЛЬНОГО ЖЕСТКОГО ДИСКА

Дискеты работают медленно. Подключите виртуальный жесткий диск — загрузка будет быстрее, и вы сможете переключаться между программами без замены дискет. *Hatari* поддерживает три типа эмулируемых жестких дисков: ACSI, IDE и GEMDOS. Из них ACSI является наиболее совместимым вариантом и работает со всеми версиями TOS от 1.02.

Эмулированные диски ACSI существуют физически как файл **IMG**, и их бывает трудно настроить, плюс они требуют драйверы, размещенные на виртуальном жестком диске для правильной работы. К счастью, *Hatari* предоставляет образ пустого жесткого диска на 80 МБ, который можно загрузить с <https://hatari.tuxfamily.org/hd80mb.image.gz>. Извлеките файл **hd80mb.image** в подходящую папку, затем откройте настройку *Hatari*, выберите Hard disks и нажмите Browse рядом с изображением ACSI HD, чтобы выбрать файл.

Перезагрузите ST, и вы увидите новый диск C, куда вы можете скопировать файлы. Учтите, что многие коммерческие игры откажутся запускаться с жесткого диска. Те, что запускаются (например, *Civilization*), вы просто держите в отдельных папках и убедитесь, что сохранили конфигурацию, чтобы жесткий диск был всегда доступен, когда вы запускаете свой виртуальный ST.

```
$ lsusb
```

Вы должны увидеть свой адаптер MIDI в списке. Теперь введите следующее (что должно проверить доступность по крайней мере одного порта MIDI):

```
$ aplaymidi -l
```

Если порты доступны, откройте *Hatari* и нажмите F12, затем нажмите Devices. Нажмите две кнопки Browse в Enable MIDI emulation и выберите для них любую запись MIDI, отображенную в `/dev/snd`. Отметьте Enable MIDI emulation и перезапустите виртуальный ST.

Если у вас нет MIDI-устройства, попробуйте использовать программный MIDI-синтезатор. Сначала установите *Virtual MIDI Piano Keyboard* (выполните поиск Pi > Preferences > Add/Remove Programs для 'vmpk'). Запустите *QSynth* из Pi > Sound & Video, а затем *VMPK*. В *VMPK* выберите Edit > Connections, нажмите Output MIDI connection и выберите FLUID Synth, чтобы связать их друг с другом. Нажмите виртуальные клавиши, и вы должны услышать звук, для проверки, что он работает правильно.

Теперь откройте окно терминала и введите следующие команды:

```
$ sudo modprobe snd-virmidi
```

```
$ aconnect -i -o
```

Запишите номера клиентов и введите

```
$ aconnect <sender> <receiver>
```

Замените `<sender>` номером клиента выхода *VMPK*, а `<receiver>` — номером клиента первого порта Virtual Raw MIDI. Теперь повторите команду, на сей раз заменив `<sender>` на порт Virtual Raw MIDI, а `<приемник>` — на входной порт Synth, например, вот так:

```
$ aconnect 128 28
```

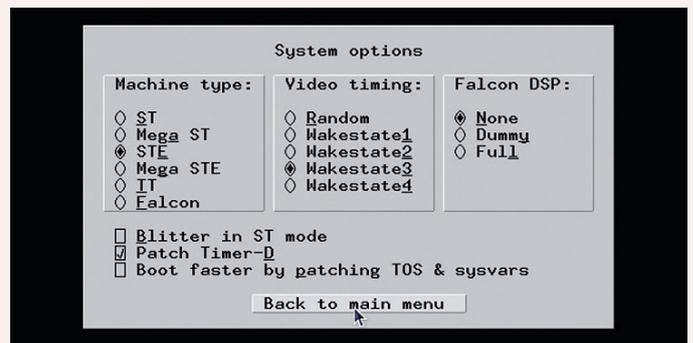
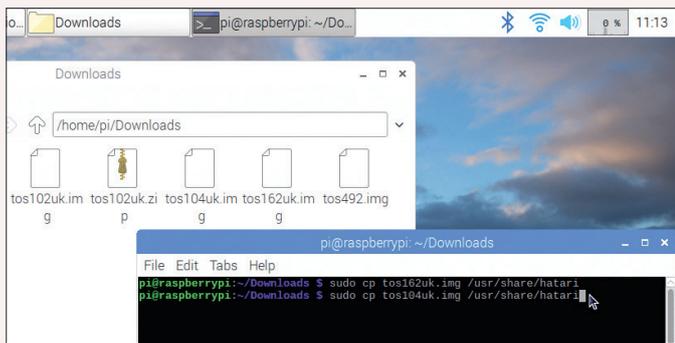
```
$ aconnect 28 130
```

Теперь откройте *Hatari*, нажмите F12 и выберите Devices, затем укажите ему правильное соединение. Отметьте Enable MIDI emulation, и вы сможете использовать свой виртуальный синтезатор для ввода музыки в программу секвенсирования типа *Cubase*, прослушивая через *Qsynth*. Готово! **LXF**

СКОРАЯ ПОМОЩЬ

Нажмите экран *Hatari* и отметьте Fullscreen, чтобы запустить его в полноэкранном режиме по умолчанию, или нажмите AltGr+F для переключения между полноэкранным и оконным режимами. Жмите AltGr+M для захвата мыши в оконном режиме.

НАСТРОЙТЕ ВИРТУАЛЬНЫЙ ATARI ST



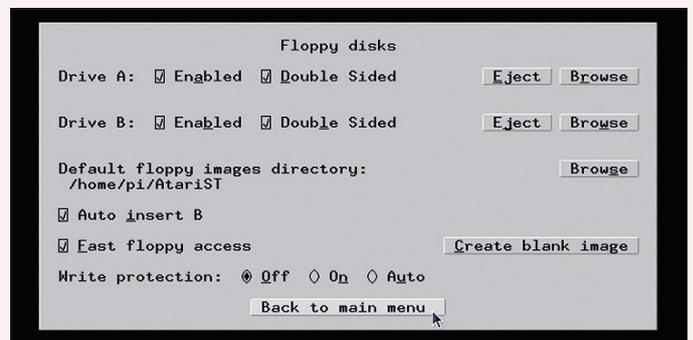
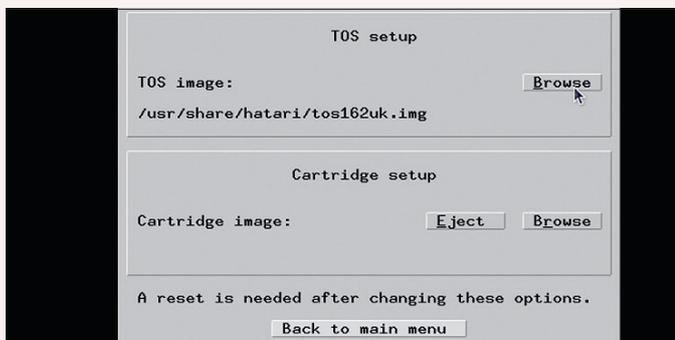
1 Скопируйте файлы TOS

Откройте File Manager. Дважды щелкните по загруженному ZIP-файлу **TOS**, чтобы открыть его в Archive Manager, затем перетащите файл `.img` в папку **Downloads**. После этого нажмите F4, чтобы открыть окно терминала, указывающее на папку **Downloads**, и введите следующее (замените `tos102uk.img` на свой файл образа):

```
$ sudo cp tos102uk.img /usr/share/hatari
```

2 Установите тип системы

Теперь откройте меню Pi и выберите Games > Hatari. Нажмите OK на экране Could not load TOS file. Когда появится главное меню, нажмите System. По умолчанию *Hatari* настроен на эмуляцию STFМ, но вы можете выбрать другую машину Atari из списка. Отметьте Boot faster, исправляя TOS & sysvars, независимо от того, какую машину вы выбираете, чтобы немного ускорить время загрузки.



3 Выберите ПЗУ и ОЗУ

Вернитесь в главное меню и нажмите ROM. Нажмите Browse, чтобы выбрать образ TOS, который вы скопировали на первом шаге — убедитесь, что он соответствует эмулируемой вами машине Atari. Нажмите Back для возврата в главное меню, затем нажмите RAM. По умолчанию выделяется 1 МБ, этого хватает для игр — уменьшите до 512 КБ для некоторых старых игр, если они откажутся загружаться.

4 Выберите дискету и сохраните настройки

Теперь нажмите Floppy disks, затем Browse рядом с каталогом **Default floppy images**, чтобы выбрать папку, в которой хранятся ваши образы дисков. Отметьте Fast floppy access, чтобы ускорить время загрузки, но обратите внимание, что некоторые диски могут не работать (в таком случае иногда помогает снятие этого флажка). Нажмите Back в главное меню, а затем Save config, чтобы сохранить настройки.

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 lxformat@futurenet.com, чтобы получить решения и предложения от знаменитого д-ра Бидвелла.

В Двухшаговый Ubuntu
У меня есть ноутбук, который я использую оффлайн с Windows 7 и Ubuntu 14.04. Помимо этих двух разделов с ОС, у меня есть два NTFS-раздела (для данных и VM). Моя 14.04 установка не обновлялась с 2016 г. Правильно ли я думаю, что я не могу apt upgrade мою установку 14.04 после выхода 18.04? Как мне без проблем обновиться до 18.04?

Я новичок, и мне не хочется форматировать раздел Linux и всё переустанавливать.

Кажется, мой Wi-Fi перестал работать с адаптивным порталом в моей библиотеке — может ли причина быть в том, что дистрибутив так долго не обновлялся?

Оливер [Oliver], через форумы LXF

О Хорошие новости, Оливер: Ubuntu 14.04 будет поддерживаться до апреля 2019 г. — релиз LTS идет с пятилетней поддержкой. Так что у Вас есть целый год на решение этой проблемы. Это не означает, что Вы ничего

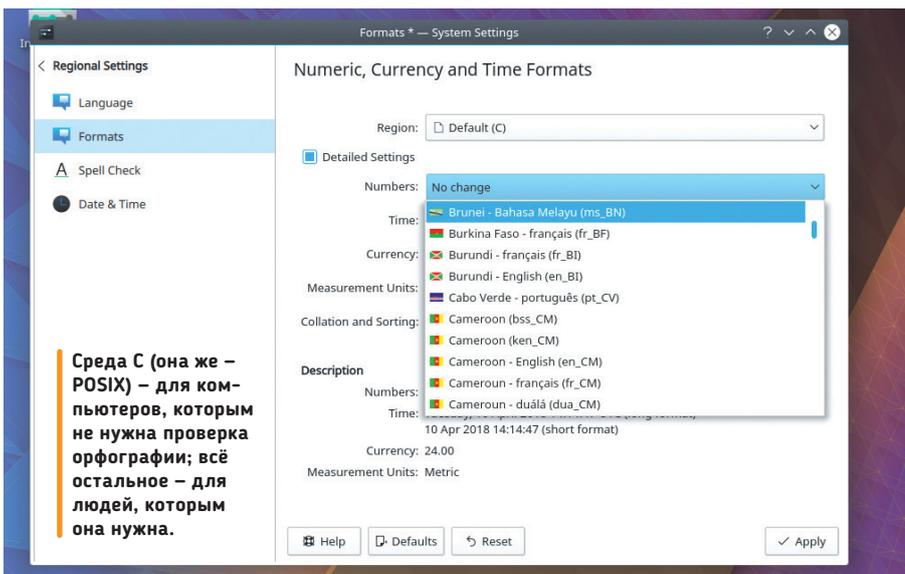
не должны делать: как минимум, Вам стоит найти проводное Ethernet-соединение и обновить свою установку 14.04. Вряд ли это что-нибудь повредит, и может даже исправить Ваши проблемы с беспроводным соединением. Больше я ничем не смогу помочь Вам с этой проблемой, особенно не зная, какое используется оборудование, но без колебаний могу сказать, что адаптивный портал Wi-Fi — ужасная вещь, и Вам стоит всячески его избегать.

Вы совершенно правы: если бы Вам пришлось делать чистую установку 18.04, надо было бы заново устанавливать все Ваши программы (помимо таких вещей, как LibreOffice и Firefox, которые установлены по умолчанию). Также велика вероятность, что некоторых из этих программ больше нет в репозиториях, и Вам придется либо компилировать и устанавливать их вручную, либо находить альтернативы.

Если Вы хотите обновиться до 18.04, то чистая установка, вероятно, будет наилучшим способом, при условии, что все нужные Вам приложения всё еще существуют. Всё, что Вам надо сделать — это резервные копии всех требуемых Вам документов из домашней папки (или откуда-либо еще на целевом диске — если у Вас много данных, пригодится внешний диск), выполнить установку (ваши NTFS-разделы можно оставить нетронутыми), восстановить Ваши зарезервированные файлы и установить все необходимые программы. Даже на старом ноутбуке (пока на нем есть место) это не должно занять больше пары часов. Однако всё редко идет по плану, так что имеет смысл отвести на это больше времени.

Альтернативный путь, который не включает очистку Вашего жесткого диска — сначала обновиться до 16.04. Возможно, Вы столкнетесь с некоторыми трудностями, но всё может пройти и гладко. Далее Вы можете либо остаться с 16.04 (и поддержкой до 2021 г.) или же обновиться до 18.04. И вновь, обновление может пройти гладко или не очень, и Вы останетесь с парой реликтов в виде не одного, а целых двух Ubuntu.

Однако, возможно, Вам и незачем обновляться до 18.04, по крайней мере, не прямо сейчас. Там используется рабочий стол Gnome, который после отказа от Unity в 14.04 потребует привыкания. Вы получите исчерпывающую информацию в нашей прекрасно написанной вводной статье. Возможно, 18.04 также не поладит со старым оборудованием. Обновление до 16.04, возможно, лучший вариант — внешне она похожа на 14.04.



Среда C (она же — POSIX) — для компьютеров, которым не нужна проверка орфографии; всё остальное — для людей, которым она нужна.

» BALL И (KEY) CHAIN

В В прошлом году я рассматривал функции разных дистрибутивов Linux и где-то услышал (DistroWatch, Linux Format или, возможно, даже YouTube) о дистрибутиве, в котором есть встроенный эквивалент Apple Mac OS Keychain; но, к сожалению, я не помню, что это за дистрибутив. Не могли бы вы мне помочь?

Чарльз Кин [Charles Keane], через электронную почту

О Не припомню, чтобы я когда-либо видел подобный дистрибутив, и не понимаю, зачем бы дистрибутиву

эмулировать инструмент Apple в Linux. Утилита Seahorse (также известная как Passwords and Keys) для Gnome прекрасно справляется с управлением ключами (GPG и SSH), паролями и сертификатами. Может, именно ее Вы и видели?

KDE Plasma имеет свой собственный KWallet, выполняющий такую же задачу, но в нем имеются ошибки, а еще, по моему опыту, он склонен забывать об автоматической разблокировке при входе в систему. Обе программы по умолчанию имеют интеграцию в Chromium, также доступны примочки для Firefox.

Если это не сработает или не устроит Вас, попробуйте чистую установку одной из облегченных разновидностей 18.04 — Lubuntu, Xubuntu или Ubuntu Mate. Однако я бы пару месяцев повременил с 18.04...

В Битва за NASерот

Я прочитал предложенный ответ на «Клуб NAS» в вашем журнале (LXF235) и подумал, что стоит тщательно изучить альтернативный ответ. Я бы предложил настроить VPN и перевести компьютер NAS в оффлайн, чтобы между другим сайтом в Германии и сайтом в Великобритании существовал VPN. Компьютер на любой из сторон, например, VPN-роутер на любой из сторон, может быть VPN типа сеть-сеть или выполнять агрессивное соединение IPSEC VPN из Германии.

Разве это не решит проблему видимости в обычном Интернете (естественно, VPN-сервер видим), и не обеспечит более прочную настройку, лучший вход в систему и лучшие опции защиты/черного списка? В той же теме вы говорите о снупинге, безопасности и т.д., поэтому я бы считал, что стоит рассмотреть такую возможность.

Некоторые домашние роутеры позволяют настроить это напрямую. Возможно, в будущих выпусках имеет смысл рассмотреть это подробнее.

Стюарт Смайлз [Stuart Smiles],
по электронной почте

О Спасибо, Стюарт. Я думал о преимуществах и недостатках некоего связанного с VPN решения (честно), но решил, что оно



Trusty Tahr был превосходен, и будет поддерживаться еще год. Но уже пора подумать об обновлении.

потребуется весьма солидного планирования. Одно дело — сказать: «Просто настройте VPN», и совсем другое — рассказать, как именно это делается.

Кроме того, мы должны подумать о технических знаниях людей на континенте. Большинство вполне устраивает возможность ввести пароль на странице HTTPS с хорошим динамическим DNS URL. Некоторых совершенно не устраивает возня с VPN-соединением с сертификатами и ключами и затем ввод загадочного

URL для доступа к NAS. Можно упростить данный процесс, но в конечном итоге это просто сервер, для доступа к которому требуется пароль, т.е. по сути совершенно то же самое, что и изначальная настройка NAS.

Но Вы правы, с технической точки зрения VPN является более предпочтительным способом. И настройку VPN мы в будущем обязательно рассмотрим. Однако работа с VPN дома не обязательно защитит вас от домашнего шпионажа, да и доверие встроенным в роутеры функциям VPN тоже может оказаться не слишком разумным. LXF

» ОРФОГРАФИЧЕСКИЕ ПРИЧУДЫ

В У меня странная проблема. При использовании LibreOffice в KDE Plasma не видна проверка орфографии. Сначала я вообразил, что у меня безупречное правописание, но над моей работой начали смеяться коллеги. Естественно, неправильно написанные слова не подчеркивались, и запустив проверку вручную, я получил сообщение «Проверка орфографии завершена», несмотря на мои ошибки. Если покопаться в опциях, я могу найти установленные словари, но в диалоговом окне Spelling [Орфография] выпадающее меню Text Language [Язык Текста] пустое. Я использую Arch, и установил всё, что считал необходимыми пакетами — hunspell, hunspell-en и hunspell-en_GB — но эффекта никакого.

Я знаю, что правильно установил язык системы (через /etc/locale.conf), что подтверждает запуск localectl в TTY, но я заметил — в сессии KDE эта команда сообщает мне, что язык настроен на C, что, конечно, неправильно. Кажется, выбор тем неправильный, хотя у меня установлена тема Breeze GTK, и другие приложения GTK выглядят хорошо. Где-то я читал о замене переменной среды

SAL_USE_VCLPLUGIN, в соответствии с Help > About. Я уже использую Qt4 VCL.

Уилл Озадаченный [Will de Bewilded], Уилтшир

О Выше нос, Уилл. Не переживайте, Ваш секрет известен только мне и нашим 20000 читателей. У меня такая же настройка и такая же проблема. Региональные настройки в Plasma общие. Вероятно, Вы уже нашли опции Language [Язык] (System Settings > Personalisation > Regional Settings > Language [Системные настройки > Персонализация > Региональные настройки > Язык]); в таком случае Вы близки к решению.

А еще в Regional Settings имеется опция Formats [Формат], которая настраивает языковую среду для рабочего стола и периодически по какой-то причине бывает установлена на C по умолчанию.

Я могу также помочь с темами LibreOffice. К сожалению, релиз 6.0 вышел слишком поздно, чтобы включить новый VCL Qt5, поэтому для пользователей Plasma Qt4 будет лучшим выбором. Хитрость в том, чтобы включить тему Breeze Qt4, которую можно настроить с помощью qtconfig-qt4.

ПОЛУЧИТЕ ПОМОЩЬ!

Мы рады ответить на все вопросы по Linux, которые вы присылаете на lxformat@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 lshw. 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

Gnome » AmaroK » Shotwell » RawTherapee » Krita
 » Krunner-translator » Heimer » Stacer » Qsoundrec » Tower
 game » Battle for Wesnoth



Александр Толстой

всё еще приходит в себя после многочасовой игры в *Tower game* (см. стр. 104). Как мы его понимаем...

СРЕДА РАБОЧЕГО СТОЛА

Gnome

Версия: 3.28 Сайт: www.gnome.org

Команда Gnome выпускает новую сборку примерно каждые полгода. Это означает, что вы можете ожидать регулярного появления новой версии 3.x. Для *HotPicks* этого месяца мы решили выделить самую последнюю и самую лучшую версию Gnome 3.28 по двум причинам. Во-первых, здесь много интересного, а во-вторых, из-за неоспоримого факта, что Gnome в последнее время сделался одной из самых популярных сред рабочего стола. Это выбор по умолчанию дистрибутивов Ubuntu и Fedora, двух безусловно главных тяжеловесов в мире Linux.

Список новых и обновленных функций весьма обширен. Одни бросаются в глаза сразу, а присутствие других станет заметно только со временем.

Исходный шрифт Cantarell подвергся генеральной чистке и теперь выглядит куда аккуратнее благодаря исправлениям в интервалах. У шрифта появились две дополнительные опции. Куда более радикальное изменение — удаление объектов рабочего стола. Хотя большинство пользователей Gnome уже отошли от привычки размещать значки на рабочем столе, всё же можно было включить эту функцию в Dconf; однако сейчас ее уже нет (используйте *Nemo* вместо *Nautilus*, если значков на рабочем столе вам не хватает). Итак, теперь всё управление файлами сосредоточено вокруг окна *Nautilus*. Нам очень понравилась новая возможность тегирования в *Nautilus*, которая теперь может отмечать файлы и папки звездочками. В дальнейшем это поможет при сортировке и поиске.

Идем дальше: нас ждет еще несколько приятных сюрпризов в Gnome 3.28, в том числе *Boxes* и *Photos*. *Boxes* теперь умеет автоматически качать ISO-образы других операционных систем, а *Gnome Photos* — импортировать файлы с SD-карт и дисков USB, а затем группировать их по альбомам. Так что в ряде сценариев больше не надо устанавливать сторонние программы, чтобы выполнить свои задачи. В частности, *Photos* сейчас предлагает базовые функции *Rapid Photo Downloader* (www.damonlynch.net/rapid/), а применение *Boxes* иногда удобнее некоторых классических решений виртуализации, таких как *VirtualBox*.

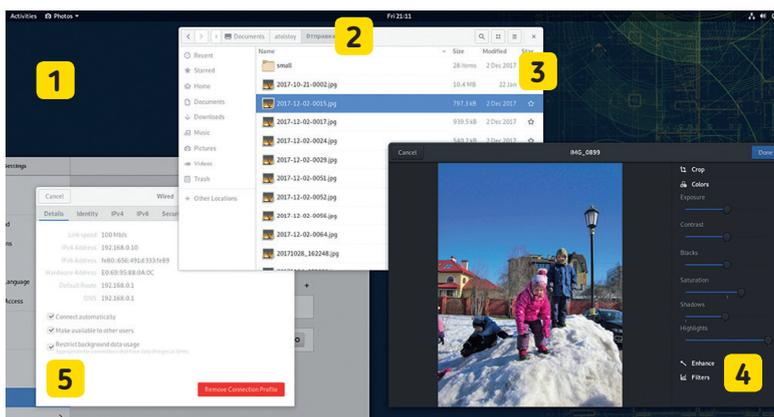
В результате, если вы любите Gnome и способы его работы, вам определенно понравится этот релиз. Нетерпеливые читатели



будут рады узнать, что прежде чем обновленные пакеты будут выпущены для вашего дистрибутива, их можно протестировать немедленно через образы live. Они основаны на openSUSE Tumbleweed или дневных сборках Ubuntu.

Сетка приложений в Gnome предлагает большие и подробные значки, приятные для глаз.

Знакомимся с интерфейсом Gnome...



1 Идеально чистый рабочий стол
 Менеджер файлов *Nautilus* больше не поддерживает значки на рабочем столе, так что обоим вашему рабочему столу будут видны во всей красе.

2 Более умный способ закрывать вкладки
 Помимо разных отладок в базе кода, *Nautilus* теперь позволяет закрывать вкладки нажатием на среднюю кнопку мыши.

3 Задавайте рейтинг файлов!
 Теперь вы можете присваивать своим файлам и папкам звезды, и использовать

эту рейтинговую систему для быстрого поиска любимых элементов.

4 Лучшие фото
 Это простое приложение предлагает достойные возможности редактирования, включая полезный инструмент улучшения светотени.

5 Учет пропускной способности
 Используйте это для отключения фоновой работы сети, что не позволит Gnome проверять наличие обновлений.

МУЗЫКАЛЬНЫЙ ПЛЕЙЕР

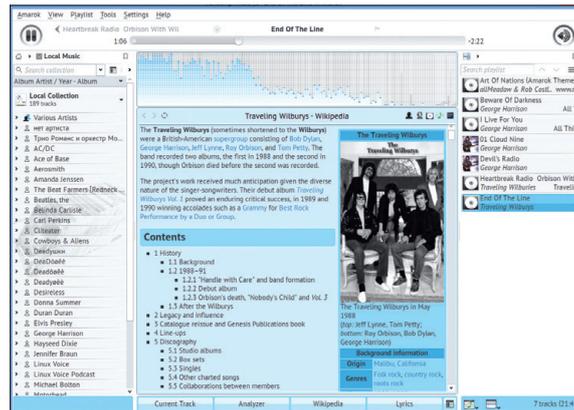
Amarok

Версия: 2.9 Сайт: <https://amarok.kde.org>

Новый релиз *Amarok* наконец-то увидел свет, спустя почти три года застоя. Те, кто в тренде классического воспроизведения музыки на рабочем столе Linux, возможно, уже знают, что на смену стареющему *Amarok*, на базе *Qt4*, давно пришел *Clementine* (ответвление более старого *Amarok 1.4*), а недавно немало шумели о новом плеере *Babe*, о котором мы рассказывали в **LXF230**.

Самой оригинальной чертой плеера *Amarok* всегда была (и остается) интеграция его многочисленных функций в одно окно типа управления миссией. Фактически, когда мы использовали *Amarok*, нам хотелось максимизировать окно, чтобы дать каждой панели и рамке больше места для размещения всего контента. Центральная область в *Amarok* — работающий на *QtWebKit* браузер, отображающий слова песни и страницу исполнителя в *Wikipedia*. Мы подумали, что подобный подход предполагает, что пользователь не будет оставлять плеер свернутым, а захочет узнать больше о треке, который «воспроизводится в данный момент».

Эта тенденция, поощряющая вас сделать нечто большее с вашей музыкой, наблюдается повсюду. *Amarok* также просит



Левая панель в *Amarok* может делать буквально всё, и даже включает менеджер файлов.

вас прикрепить теги к музыкальным файлам (выбрать еще один трек и найти редактор тегов в текстовом меню) и найти медиа-источники онлайн, такие как *Last.fm* или *Magnatune*. Помимо того факта, что *Amarok* справился бы со всем, чего не умеет, скажем, *Clementine*, нам очень понравились знакомый броский прогресс-индикатор *Amarok*, необычный вращающийся элемент регулировки громкости и приятная для глаз анимация анализатора.

В версии 2.9 исправлены десятки ошибок с 2015 г., и она выводит плеер в современный мир, позволяя компиляцию и сосуществование с другими музыкальными плеерами. Так, *Amarok* теперь полностью совместим с *Ffmpeg 3* и поддерживает синхронизацию музыкальной библиотеки с *Rhythmbox*, *Clementine*, *Banshee* и даже с *Apple iTunes*. При такой превосходной совместимости, почему бы не дать *Amarok* шанс на вашем рабочем столе?

ФОТОРЕДАКТОР

RawTherapee

Версия: 5.4 Сайт: www.rawtherapee.com

R*awTherapee* предназначен для обработки RAW-файлов, созданных зеркальной камерой. В настройках таких камер есть опция указания формата сохраненных файлов, и вместо стандартного JPEG можно выбрать RAW. Эти файлы занимают больше места на диске, но содержат значительно больше информации, полученной с сенсора камеры. Короче говоря, они похожи на негатив фотографии, только в цифровой форме. *RawTherapee* может придать смысл данным внутри RAW-файлов: настроить уровни цвета, исправить проблемы с экспозицией, сохранить детали в тени и на свету, удалить шум и некоторые другие обычные дефекты фото.

Обычно считается, что в руках тех, кто знает, что делает, вручную обработанный RAW файл будет иметь лучшее качество, чем JPEG, сляпанный камерой в автоматическом режиме. Интерфейс *RawTherapee* темен (как темная комната для работы с пленкой), и в нем множество регуляторов, вкладок и кнопок. Но как только вы к нему привыкнете, вы поймете, что основных режимов у приложения два: браузер изображений и редактор.

Дважды щелкните по изображению в браузере и подождите секунду, пока оно загрузится. Обычно RAW-изображения бледные, и надо вручную применять цветовую коррекцию, чтобы они выглядели ярче и живее. К счастью, в *RawTherapee* имеется волшебная кнопка *Auto levels* прямо на вкладке *Exposure* [Выдержка],



с которой вы начинаете по умолчанию (см. правую панель рядом с изображением). У других вкладок есть собственные наборы кнопок управления для тонкой настройки деталей, например, резкости, баланса белого, *retinex*, геометрии и т.д. Конечно, чтобы добиться хороших результатов, нужны время и умение. Но это дело стоящее, поскольку все лучшие, получающие награды изображения, которыми мы восхищаемся на многих онлайн-конкурсах фотографий, созданы с помощью программ обработки RAW.

RawTherapee поддерживает большинство проприетарных форматов RAW, включая *Pentax*, *Sony Pixel Shift* и *Canon Dual-Pixel*. Если вы не уверены, вы всегда можете попробовать открыть файлы со своей камеры с помощью *RawTherapee* и посмотреть, работает ли это (должно!).

Вот, мы сделали розы ярче, не утратив мелких деталей в гребках.

МЕНЕДЖЕР ФОТО

Shotwell

Версия: 0.28

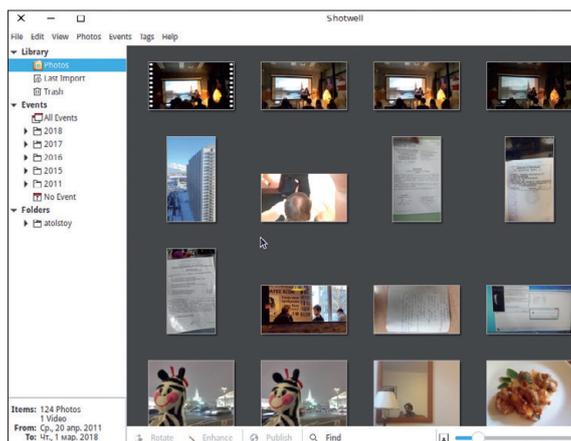
Сайт: <https://wiki.gnome.org/Apps/Shotwell>

Xотя очень трудно закрыть глаза на быструю разработку *Photos* от GNOME, есть еще более мощный менеджер коллекции фотографий. Самая последняя версия *Shotwell* появилась на полгода позднее запланированного, но наконец-то она здесь, и она превосходна!

В *Shotwell* не только исправлено более 60 ошибок — приложение теперь работает более эффективно. В нем имеется ускоренный плагин импорта RAW, лучшая поддержка изображений с альфа-каналом (прозрачность) и новый полноэкранный режим для просмотрщика изображений *Shotwell*.

Конечно, весь комплект не исчерпывается просмотрщиком изображений. Вы оцените мощные функции программы, дойдя до импорта снимков с вашей камеры или смартфона, особенно если хотите организовать свой файлы по альбомам на основе базы данных, добавить теги и события и присвоить изображениям рейтинг с помощью звезд. Все эти метаданные будут весьма кстати по мере роста вашей библиотеки — вам будет куда проще осуществлять навигацию по своей солидной коллекции фото.

Помимо функций каталога, в *Shotwell* также имеется простой редактор изображений. Для *GIMP* он не соперник, но вполне удовлетворит скромные потребности среднестатистического



Shotwell автоматически организует вашу библиотеку с помощью событий. Структура папок также остается той же самой.

пользователя. Выберите изображение в *Shotwell* и обратитесь к кнопкам внизу, чтобы повернуть, обрезать его, выровнять горизонт и удалить эффект красных глаз. Здесь есть и волшебная кнопка «исправить всё» — только она называется Auto Correction [Автоисправление], возможно, чтобы немного занизить ваши ожидания. Она не сможет восстановить тени и свет, но достаточно хороша для более простого исправления в стиле auto levels.

Shotwell теперь поддерживает систему сборки Meson, на которую недавно перешел проект GNOME. Однако вы можете скомпилировать *Shotwell* с помощью классической последовательности `./configure && make && sudo make install`. В отличие от многих других приложений, зависящих от других суперсовременных компонентов, которых у вас может не быть, *Shotwell* отличается толерантностью и позволяет быстро скомпилировать свой код с очень малым числом зависимостей.

РЕДАКТОР ИЗОБРАЖЕНИЙ

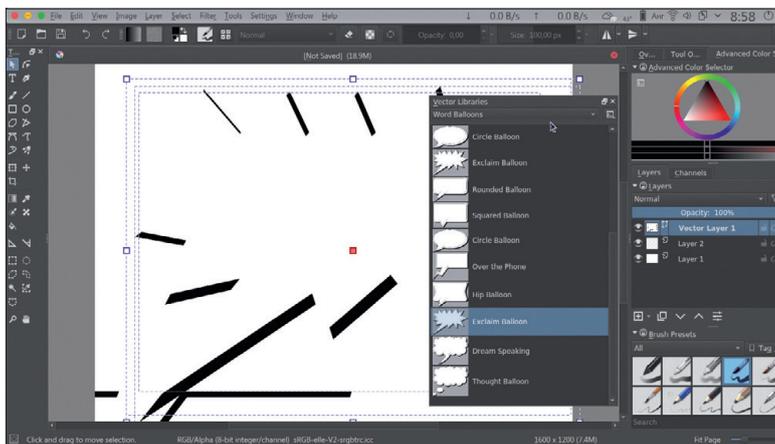
Krita

Версия: 4.0 Сайт: www.krita.org

Krita давно считается соперником *GIMP*, однако со временем программа выросла в нечто большее и, по мнению некоторых, более эффективное. Мы уже рассматривали ранее *Krita* в **LXF217** и **LXF227**, но с тех пор в это приложение многое добавилось. И всё благодаря щедрости тех, кто поддерживает *Krita* и финансирует его быструю разработку.

Krita 4.0 — смелый релиз со множеством новых функций. К обычным инструментам программы для редактирования растровых изображений теперь прибавились векторные. Более того, *Krita* перешла с формата ODG на SVG и стала полностью совместима с *Inkscape*. А значит, вы можете копировать и вставлять векторные рисунки и фигуры из одного приложения в другое без всяких проблем. *Krita* позволяет заполнить и создать контуры любой формы, выбрать ширину и стиль штриха, объединить и нахлестнуть фигуры, повернуть и изменить размер всего. Вам больше не нужен отдельный векторный редактор для выполнения основных задач по созданию векторных рисунков.

Как и следует ожидать, *Krita* идет с отличными наборами векторных объектов, идеально подходящих для создания мультфильмов и комиксов (нам очень понравились новые облачка для речей!) и новые инструменты для ускорения процесса раскраски



в комиксах. Новый инструмент Colorize Mask позволяет в черновике отметить части вашего рисунка цветом. Затем ваше произведение раскрашивается автоматически с удивительной точностью.

Далее, кисти программы по умолчанию подверглись тщательной модернизации — и, конечно, вы можете настроить и собственные кисти. В **LXF227** мы рассматривали *G'MIC*, который позволяет создавать разные эффекты в изображении. *Krita 4.0* теперь интегрируется с *G'MIC* очень хорошо, так что больше не надо подсоединять его отдельно. Вообще-то *G'MIC* дает пользователям *Krita* доступ к десяткам художественных и творческих эффектов, и у вас уйдет уйма времени даже на то, чтобы все их просмотреть.

Запуск самого последнего релиза *Krita* производится через пакет AppImage, доступный на сайте проекта.

Страстно желаете создать комикс? *Krita* поможет реализовать эту амбицию.

РАСШИРЕНИЕ PLASMA

KRunner-translator

Версия: 1.0

Сайт: <https://github.com/naraesk/krunner-translator>

Возможно, вас удивит тот факт, что количество отдельных приложений для перевода ограничено. Даже если мы говорим об интерфейсе для популярных открытых сервисов типа Google, Yandex или Baidu, есть лишь несколько достойных программ, которые мы можем рекомендовать хоть с какой-то степенью уверенности.

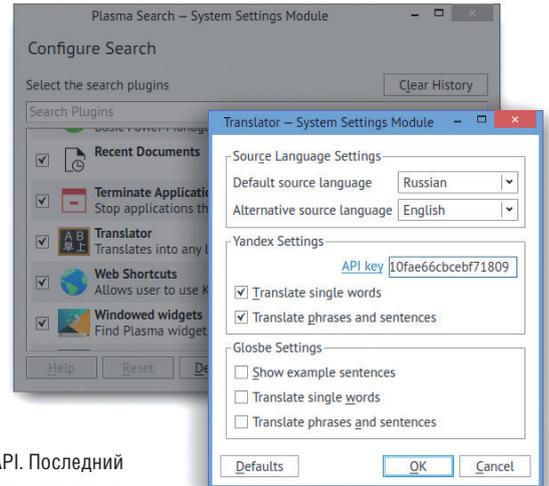
Одна из них — *KRunner-translator*, расширение Plasma 5, которое добавляет функций к вашему рабочему столу KDE Plasma. *KRunner* — это поистине универсальный инструмент, у которого множество мощных функций скрыты за простой строкой поиска (доступной через Alt+F2).

KRunner может искать файлы, места, историю просмотров, переключать сессии пользователей, делать вычисления и много чего еще благодаря своей простой природе плагина. *KRunner-translator* добавляет возможности перевода на разные языки, которые хорошо работают, если вам надо перевести фразу или быстро добыть значение незнакомого слова на иностранном языке. Вот как он работает.

Как только вы установите плагин через стандартную последовательность `$ cmake . && make && sudo make install`, найдите в настройках *KRunner* новый пункт *Translator*. Чтобы расширение

заработало, нужна предварительная настройка. Плагин поддерживает многоязычный онлайн-словарь Glosbe и Yandex Translator API. Последний начнет работать, как только вы создадите учетку в Yandex и получите ключ API, который затем надо будет вставить в окно настройки *KRunner-translator*. Включите необходимые флажки, и всё будет готово к работе.

Когда в следующий раз вам понадобится что-то перевести, используйте в окне ввода *KRunner* следующий синтаксис: <код целевого языка><исходная строка>. Например, введя 'de knowledge transfer', в ответ вы получите нечто вроде 'Wissensvermittlung'. Не забудьте указать в настройках плагина язык источника по умолчанию. Также можно выбрать первичный и вторичный язык источника, что обеспечит правильный перевод вашего поля ввода *KRunner*, и сообщить плагину, на какой язык вы хотите перевести фразу, с помощью следующего обозначения: 'fr-en Au reservoir', например.



На настройку у вас уйдет пара минут, после чего вы сможете делать перевод прямо в строке поиска.

СОЗДАНИЕ ИНТЕЛЛЕКТ-КАРТЫ

Heimer

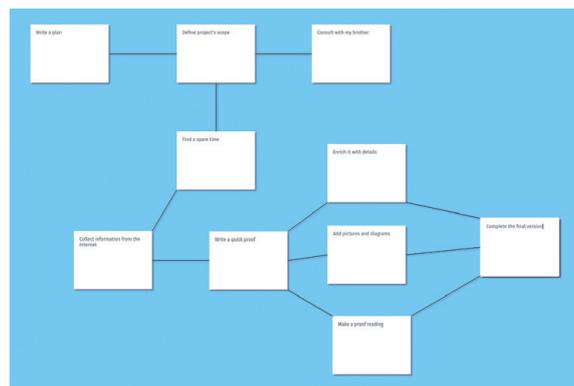
Версия: 0.0.0

Сайт: <https://github.com/juzzlin/Heimer>

Быстрый поиск программ интеллект-карт в обычных точках открытого кода показал, что имеется более 900 проектов с такими функциями; но есть вероятность, что обычный пользователь Linux даже не слышал о большинстве из них. Это, конечно, не беда: мир компьютерных технологий XXI века полон интересного, а количество часов в сутках ограничено, и вполне естественно, что есть программы и инструменты, ускользнувшие из нашего поля зрения.

Вероятно, это достаточная причина познакомиться с *Heimer*, простой программой, которая позволит вам как-то упорядочить свои идеи, объекты и концепции, поместить их в контекст и показать их взаимоотношения с другими.

В частности, *Heimer* создает иерархические диаграммы, четко показывающие отношения между элементами внутри вашей подборки объектов. Инструмент находится на ранней стадии (что объясняет его версию 0.0.0 на момент написания), но он весьма эффективно всё упорядочивает. Создав новый проект (File > New [Файл > Новый]), вы увидите один ярлык с полем ввода текста внутри, как в наклейке рабочего стола. Можете поместить в него любой текст, а закончив, просто нажмите большой значок '+' внизу, чтобы создать дочернюю заметку. Таким образом легко



Это как если поместить небольшую часть MS Visio в совершенно новое приложение.

создать дерево заметок, и более того, у каждой заметки может быть несколько нисходящих, а не одна.

Чтобы ваша диаграмма выглядела аккуратно, вы можете перетаскивать заметки по листу и организовывать их в удобном для чтения порядке. *Heimer* может сохранить ваши проекты в собственном формате .dem (на базе обычного XML), и на этом, собственно и всё. Со временем добавятся другие функции редактирования, но на данный момент программа уже может подготовить узлы вашего проекта и дать вам ясное представление об элементах, которые вы решили визуализировать. Отсутствие расширенных функций можно считать достоинством: вам ничто не будет мешать организовать элементы вашего дерева, поэтому вы можете использовать *Heimer* для записи идей и для быстрых заметок.

БРАУЗЕРНАЯ ИГРА

Tower game

Версия: GIT

Сайт: https://github.com/bmqb/tower_game

Опыт быстро учит нас, как сочетаются разные вещи, и в случае с играми с открытым кодом на Github это означает, что самая частая пара — игровой движок с открытым кодом и подборки проприетарных игр на его основе. По крайней мере, именно к такому выводу вы придете, исследовав самые популярные результаты поиска.

Хорошая свободная сетевая игра, способная работать на настольном ПК — тоже большая редкость, поскольку большинство таких игр созданы для Android и iOS. Тем не менее, мы нашли чудесную захватывающую игру с весьма приземленным названием *Tower game*. Это китайский проект с файлом **Readme.md**, который надо перевести, если вам нужна информация по процессу установки, и в самой игре тоже имеются китайские слова. Однако это не создает проблем, потому что *Tower game* — это демо-страница онлайн, и играть в нее — проще простого.

Ваша цель — построить башню из обычных блоков. Каждый блок пристраивается к месту раскачиванием троса крана, и как только вы нажмете на левую кнопку мыши, кран отпустит блок. Если блок приземлится на предыдущий блок, не упав, вы переходите на следующий этаж. Чем больше вы постройте этажей,



Я на самом верху — слишком устал, чтобы зажигать...

тем сильнее будет раскачиваться кран, и становится трудно определить наилучшее время укладки следующего блока. Если вы правильно сложили предыдущий этаж, игра даст вам бонусные очки; несколько точных попаданий также украсят ваш счет. Игра не ограничена по времени, так что вы можете сколько угодно дожидаться оптимальных моментов!

Конечно, такие плохо сконструированные башни не существуют в реальной жизни. И ни одно здание не бывает достаточно высоким, чтобы достичь планет из сыра. Однако эти глупости только добавляют удовольствия от игры. Несмотря на очевидную простоту, игра очень захватывает, и вам захочется в нее играть — и чтобы побить собственный рекорд, и чтобы посоревноваться с друзьями.

СТРАТЕГИЯ

Battle for Wesnoth

Версия: 1.13.12 Сайт: www.wesnoth.org

В*attle for Wesnoth* — классическая пошаговая стратегия, в которой вы играете за вымышленную расу и сражаетесь с вражескими армиями. Игра находится где-то между *Heroes of Might and Magic* и *Age of Wonders*, хотя вам не обязательно знать эти игры, чтобы влюбиться в волшебный мир Веснота [Wesnoth] и Большого Континента [Great Continent].

Игра предлагает одиночные кампании и сетевые многопользовательские матчи. В каждом случае ваша цель — создать мощную армию и захватить деревни, производящие золото. Игра в меньшей степени экономическая и в большей — военная, поэтому золото — единственный имеющийся ресурс, который вам нужен, чтобы купить больше боевых соединений и платить им жалование.

В *Battle for Wesnoth* шесть рас: люди, эльфы, орки, гномы, нежить и драконы, и семь промежуточных рас. У каждой свои сильные и слабые стороны, которые вы можете использовать в своей стратегии. Вы ведете свои сражения на карте из шестиугольных плиток. Ваша армия состоит из разных боевых соединений и предводителя. Вы можете считать ее своего рода шахматами, только с цветными орками, эльфами и гномами вместо монохромных фигур. Каждое соединение может продвигаться по карьерным лестницам и становиться больше и сильнее.



Даже и без сражающихся боевых соединений, живописная карта так и тянет на исследование.

Это определенно способствует длительной привлекательности игры. На самом деле, *Battle for Wesnoth* существует с 2003 г. и активно поддерживается. Мы надолго потерялись в мире Веснота, и до сих пор не можем уйти из него благодаря сбалансированному сюжету и привлекательной графике. Как знать, может, мы останемся там навсегда!

ЧИСТИЛЬЩИК СИСТЕМЫ

Stacer

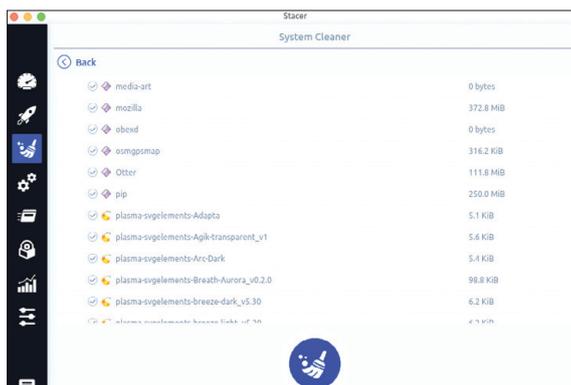
Версия: 1.0.9

Сайт: <https://github.com/oguzhaninan/Stacer>

В этом месяце *HotPicks* возглавил Gnome 3.28, с новым приложением мониторинга под названием *Usage*. Расширенных функций в нем не очень много, поэтому мы нашли более эффективную программу для мониторинга и очистки системы: *Stacer*. Официальная страница проекта говорит, что это оптимизатор для Ubuntu, но мы смогли запустить и использовать *Stacer* на нескольких других системах Linux, не на базе Debian, и он работал отлично.

Stacer — стильная настольная программа на основе Electron, которая имеет почти такой же дизайн, как *Etcher*, инструмент заливки ISO-образов, о котором мы рассказывали в **LXF213**. Оба используют темные цвета по умолчанию и предлагают простые в установке пакеты AppImage, делающие их переносимыми. Однако скромный инструмент заливки на флешку USB весит почти 80 МБ, и на его запуск уходит как минимум пять секунд, тогда как полнофункциональный *Stacer* весит всего 29 МБ и запускается заметно быстрее.

Панель управления *Stacer* отображает использование CPU, памяти и жесткого диска, а вдоль левого края окна размещает столбец монохромных значков. Помимо панели управления,



есть еще шесть разделов, позволяющих контролировать пункты автозапуска, включать или выключать сервисы, работающие на *systemd*, останавливать процессы, управлять установленными программными пакетами и, конечно, чистить файлы кэша и прочий системный мусор.

Если ваш дистрибутив Linux не использует менеджер пакетов *Apt*, то *Stacer* не сможет управлять вашими программами, а луддитам *Systemd* будет также не хватать функции менеджера сервисов.

Самой интересной частью *Stacer* был его инструмент System Cleaner, который работает с кэшем пакетов, отчетами о сбоях, журналом приложений, кэшем и корзиной. Очистка кэша приложения чревата опасностью, так как *Stacer* просто проходит по вашим dot-файлам и по умолчанию хочет удалить всё. К счастью, *Stacer* можно нацелить на удаление только заданных пунктов, и такой выборочный подход отлично работает.

О нет, не удаляйте наши драгоценные модули Python — пожалуйста!

ПРОГРАММА ЗАПИСИ МУЗЫКИ

Qsoundrec

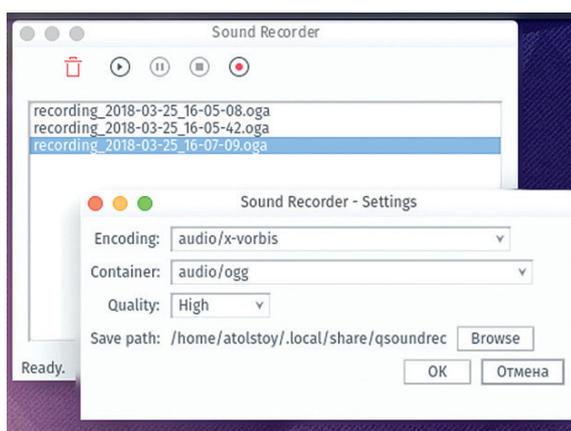
Версия: GIT

Сайт: <https://github.com/vfspirit/qsoundrec>

М ногим читателям знакома необходимость применять тяжеловесные программы для выполнения простейших задач: например, обрезать фотографии в *GIMP* или перекодировать видео из одного формата в другой через *Kdenlive*. Очевидно, и при работе с аудиофайлами мы тоже сталкивались с подобной ситуацией.

Несмотря даже на то, что Linux предлагает достаточно приложений, позволяющих записывать звук с микрофона, объединять несколько треков в одну композицию, добавлять эффекты и устранять шум, нам часто нужна программа попроще, способная записать звук с микрофона и сохранить его в выбранном вами аудиоформате.

Так, например, рабочий стол Gnome по умолчанию предлагает простой фонограф; но затем мы наткнулись на *Qsoundrec*, тоже понятный, элегантный, исключительно простой в работе и подходящий для любой операционной системы благодаря кросс-платформенной библиотеке *Qt*. Это, вероятно, главное достоинство *Qsoundrec* — без сомнения, удобного для всех, кому нужно надежное приложение на любой рабочей станции независимо от ОС. Интерфейс *Qsoundrec* прост: в главной части окна находится список ваших аудиофайлов, а наверху — ряд кнопок управления воспроизведением вместе с отдельной кнопкой



Qsoundrec хранит записи внутри вашей домашней директории, и их не удалить случайно!

Remove Record [Удалить Запись]. Всё очень похоже на хорошо продуманное мобильное приложение звукозаписи, хотя *Qsoundrec* демонстрирует классический настольный дизайн.

Меню Edit > Settings [Редактировать > Настройки] содержит ряд основных предпочтений, которые можно изменить: аудиокодек, его контейнер и уровень сжатия аудио (качество на выходе). По умолчанию используется бесплатный кодек Ogg Vorbis, но есть также множество других опций. Источник звука определяется вашими настройками *Pulseaudio*, которые можно изменить разными способами, в зависимости от вашего рабочего стола. Например, при работе в KDE Plasma *Qsoundrec* использует текущие настройки Phonon. **LXF**

На диске

Дистрибутивы, приложения, игры, книги и всякое-разное...



НЕЙЛ
БОТВИК

Переход с Windows или MacOS на сперва кажущийся загадочным мир Linux (а в итоге он оказывается интересным и полным) бывает труден, как и выбор лучшего дистрибутива для этого.

Создание дистрибутива для этой ситуации изобилует препятствиями. Позвольте мне объяснить...

Я думаю, вы согласитесь, что бессмыслен дистрибутив, который притворяется не тем, чем является. Linux и Windows — два совершенно разных монстра компьютерного мира, и новый пользователь должен с этим смириться. В равной мере нет смысла заставлять дистрибутив вести себя, как клон Windows, ибо это — тупик. Пользователь пока не знает Linux, ну разве что считает, что тот похож на Windows, и при попытке взять другой дистрибутив он испытает тот самый шок, которого мы изначально пытались избежать.

Идеальная ситуация — если есть друг, который разбирается в Linux и может познакомить новичка с разными средами (хотя так можно лишиться друга... но это так, к слову). Тогда идеальным дистрибутивом будет тот, который опытный друг хорошо знает и постоянно использует.

И хорошая штука — дистрибутив, прилагающий максимум усилий по облегчению перехода, всё же делая этот переход, чтобы новый пользователь не вечно сидел в новичках, а набирал опыт. Надеемся, кто-нибудь сумеет помочь нашим друзьям пересечь пропасть между Windows и Linux.

КРОШЕЧНЫЙ И ПОРТАТИВНЫЙ

Slax 9.4.0

32- и 64-битный

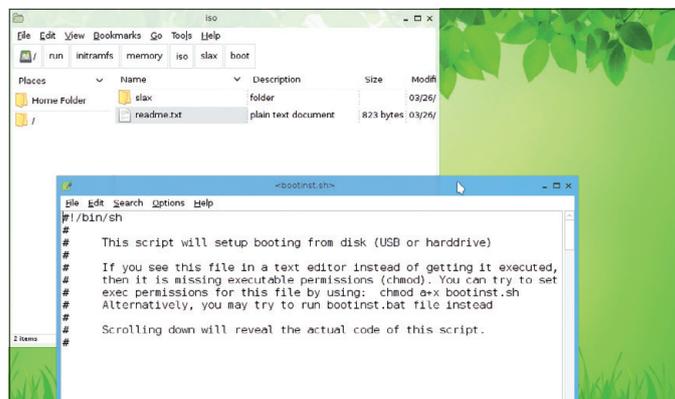
Slax — легковесный портативный дистрибутив, предназначенный для работы с флешки USB.

Мы знаем, что вы думаете: в этом нет ничего необычного, ведь почти все дистрибутивы на наших DVD можно скопировать на флешку с помощью `dd` или `Etcher`. И да, вы можете сделать то же самое со Slax; но недостатком подобного метода будет то, что он занимает всю флешку целиком — неважно, насколько компактный дистрибутив и емкая флешка.

Если вы воспользуетесь методом Slax для его установки на флешку, остальная часть флешки останется свободной для использования, и в очень удобном виде. Slax может применять сохраняемый файл или директорию для хранения файлов и настроек. Это означает, что когда вы в следующий раз загрузитесь с флешки, у вас будет всё, что вы сохранили в прошлый раз, даже если прошлый раз был на другом компьютере.

Всё, что вам надо сделать — загрузить Slax с DVD, или скопировать на флешку USB, вставить отдельную флешку и перейти в директорию `/run/intrams/memory/iso` в live-системе Slax. Здесь вы найдете директорию с именем `slax` — скопируйте ее в корневую директорию целевой флешки. Затем откройте терминал и перейдите в вашу новую директорию `slax`. Отсюда запустите следующий код:

```
$ cd boot
$ sh bootinst.sh
```



Легковесный и портативный, но со всем необходимым; Slax — это компьютер у вас в кармане.

Это делает флешку USB загружаемой; теперь можете перезагрузиться и выбрать опцию USB в меню загрузки вашего компьютера. Если вы отформатировали флешку в файловой системе FAT (или не форматировали ее, потому что она уже была в этом формате), ваши изменения будут сохранены в файле с именем `slax/changes/changes.dat`. Из-за ограничений файловой системы FAT этот файл не должен превышать 4 Гб, так что вы можете сохранить только такой объем данных. Если вы форматируете флешку в родной файловой системе Linux, например, `ext4` (хотя некоторые считают, что для флешек лучше `ext2`), то изменения будут сохранены в директории `slax/changes`, и единственное ограничение на объем хранения — емкость вашей флешки.

Приятный побочный эффект компактности Slax в том, что мы смогли уместить на DVD и 32-, и 64-битную версии.

Скопируйте одну папку и запустите этот скрипт, и у вас будет USB-флешка Slax live, сохраняющая файлы и настройки между запусками.

» 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.



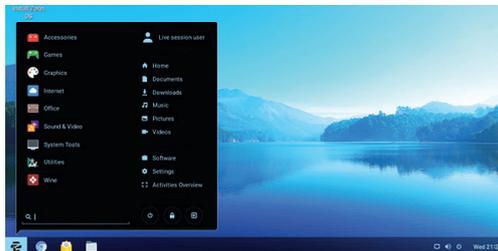
ИДЕАЛЕН ДЛЯ ПОЛЬЗОВАТЕЛЕЙ WINDOWS И MACOS

Zorin OS 12.3

64-БИТНЫЙ

Начиная в Linux, пользователи могут быть обескуражены, ведь всё настолько непохоже (естественно: в том и суть). A Zorin OS пытается сделать переход как можно менее болезненным.

Этот дистрибутив не притворяется другой ОС, он просто смягчает переход. Это важное отличие, поскольку оно означает, что Zorin — всё же «нормальный» дистрибутив Linux, который вы можете использовать, только начиная знакомство с ОС, и продолжать использовать, обретая уверенность и, может быть, став более требовательным — дистрибутив, который подходит и бабушкам, и технарям!



С ним проще смириться пользователям Windows, и тем не менее это настоящая Linux ОС. Zorin OS старается дать лучшее от обоих миров.

БИОМЕХАНИЧЕСКАЯ ЛОШАДКА ДИСТРИБУТИВОВ

Ubuntu 18.04 'RC'

64-БИТНЫЙ

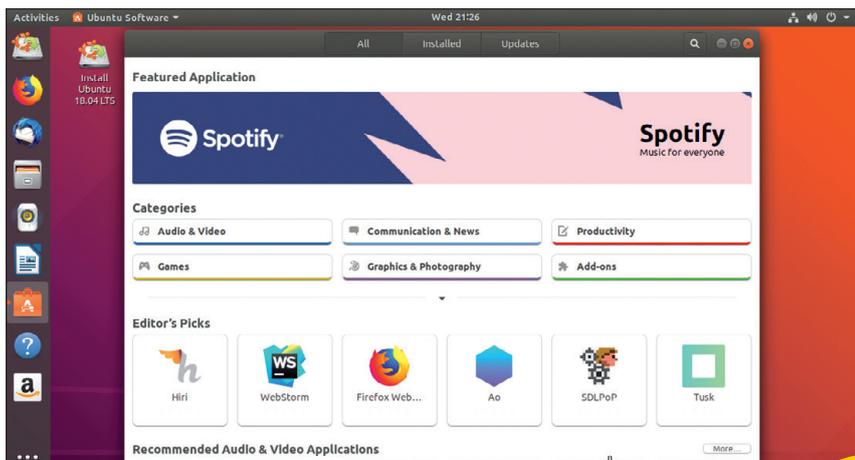
Пришло время для всеми (ну почти) любимого дистрибутива. К сожалению, производственный цикл Ubuntu на сей раз сильно разошелся с нашим, и финальный релиз вышел слишком поздно для DVD, поэтому мы использовали период заморозки ядра предвыпускной версии.

Обычно мы так не поступаем, но данная версия очень близка к финальной, и всё, чем они, вероятно, будут отличаться — несколько обновлений перевода и, возможно, позднейшая отладка.

Естественно, всё это будет доступно в виде обновлений, и после первого же обновления вы получите полную версию релиза.

Ubuntu 18.04 — это релиз LTS (Long Term Support — с долгосрочной поддержкой). Обычные релизы поддерживаются только девять месяцев, и обновления перестают выходить вскоре после выхода новой версии, но релизы LTS предлагают пять лет поддержки. А значит, вы будете получать отладки и обновления системы безопасности до 2023 г.

Однако это не значит, что вы будете получать обновления до новейших программ, поскольку LTS считается стабильным релизом. Это определение стабильности от Debian, согласно чему всё должно оставаться практически одинаковым, без сюрпризов. Но это не делает другие релизы нестабильными!



Максимально близко к финальному релизу Ubuntu 18.04; это второй релиз с рабочим столом Gnome по умолчанию.

» 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!

СПРАВОЧНИК ПО GNU EMACS

Риску отпугнуть половину наших читателей, заявив, что я редактирую тексты в Emacs, и огорчить большинство остальных, заявив, что я использую его только для редактирования текста. Emacs — невероятно мощная платформа на базе текстового редактора, который умеет много больше, чем просто редактировать текст. Поэтому и справочник к нему превышает 600 страниц. Если вы когда-либо дивились, из-за чего столько шума, можете это выяснить. От загадочных, но и логичных команд клавиатуры до макросов и даже чтения электронной почты — всё это здесь.

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** Начинать 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.
- **Нырните в Python** Всё, что вам надо знать об этом языке программирования.
- **Введение в Linux** Удобное руководство с массой подсказок для начинающих.
- **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 сисадмина Linux** Контролируйте свою систему.
- **Обзор инструментов** Полный обзор инструментов GNU.
- **Справочник GNU Emacs** Шестсот страниц информации!
- **Создание программ с открытым кодом** Всё, что вам надо знать.
- **Программирование с основ** Сделайте первые шаги.

ПРОШЛЫЕ НОМЕРА » ПРОПУСТИЛИ?

LXF236

Май 2018



- Роботы на Pi Терминатор своими руками
- Веселый архивариус Управляем хранилищем
- Сайтострой Стилизуем под кого-то
- Raspberry Pi греет Отопление на контроле

LXFDVD

FArch, ArchLabs, Arch Linux ARM, Manjaro, подборка для NAS, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_236/
PDF-версия: shop.linuxformat.ru/elxf_236/

LXF235

Апрель 2018



- Отгоним шпионов Все виды самозащиты
- Как стать телеведущим Видеоконференции
- Linux – фотографам 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 легковесы Лебединая песня?
- Gnome 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 хочю
- 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/

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

Slax 9.4.0 (32- и 64-битный)
Современная, портятивная, небольшая и быстрая ОС Linux с модульным подходом и выдающимся дизайном. Работает прямо с USB-накопителя без установки, так что ее можно носить в кармане. Несмотря на малый объем, предоставляет приятный графический интерфейс пользователя, но только две программы: веб-браузер и терминал.

Ubuntu 18.04

Поставляется со всем необходимым для работы организации, школы, дома или предприятия. Предварительно установлены официальный пакет, браузеры, электронная почта и медиа-приложения.

Zorin OS 12.3

Альтернатива Windows и macOS, разработанная для того, чтобы сделать ваш компьютер быстрее, мощнее и безопаснее.

NOTPICKS

Апгрейд 2.9 Музыкальный плеер audacious.kde.org

Книга 4.0

Редактор изображений

www.krita.org

Книга 4.0 Редактор изображений

» 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

Описание на обороте >

LINUX FORMAT

LINUX FORMAT 18.04

РЕЛИЗ-КАНДИДАТ

Релиз года, без которого буквально не жить!

- Более быстрая и легковесная сборка
- Новый рабочий стол GNOME
- Долгосрочная поддержка
- Обновленное ядро 4.15

Zorin OS 12.3

Идеальное приращение
для пользователей
Windows и MacOS

64-битный

Slax 9.4.0

Крошечный и мощный:
эту ОС можно
запустить с USB-брелка

64- и 32-битный

Содержание

ГЛАВНОЕ

Grub
Mentest86+

ДОКУМЕНТАЦИЯ: 12 книг о Linux (на английском языке)

КНИГА МЕСЯЦА: GNU Emacs Manual Руководство по GNU Emacs

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] «Соброр и базар»

The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian

Dive Into Python Учебник по программированию на Python

Пожалуйста, перед использованием данного диска ознакомьтесь с инструкцией, опубликованной в журнале на стр. 109!

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 — Греггом Кроа-Хартманом [Greg Kroah-Hartman]

Producing OSS Как выпустится СПО

Programming from the Ground Up Руководство по базовому администрированию Linux

GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

Страница 2

ДИСТРИБУТИВЫ

Рhinux OS 18.04 LTS (64-битный)
Дистрибутив на базе Ubuntu для начинающих пользователей Linux. Оснащен многочисленными улучшениями удобства, встроенной поддержкой мультимедиа-кодеков и плагинов браузера, интерфейсом GNOME с расширенным меню, панелями и док-панелями и тщательным выбором популярных настольных приложений для многих распространенных вычислительных задач.

Linux Lite

(64-битный)
Дружелюбный к новичкам дистрибутив на базе Ubuntu LTS, с рабочим столом Xfce. В первую очередь ориентирован на мигрантов с Windows и призван обеспечить полный набор приложений в помощь повседневным потребностям пользователей, в т.ч. полный оффисный пакет, медиа-плееры и др. необходимое ПО.

Rosa R10 Xfce (64-битный)
Неофициальный образ от сообщества российской ОС ROSA Линукс Фреш R10 с рабочим столом Xfce (версия образа 1.1). Легкий вес Xfce позволяет установить систему и комфортно работать на компьютерах с небольшим объемом оперативной памяти и не самым передовым процессором.

Sculpit TC-18.05 Pre-built

(64-битный)
Система общего назначения с открытым исходным кодом. Комбинирует архитектуру микроядра разработки Белоруссии, безопасность на основе возможностей, изолированных драйверы устройств и виртуальные машины для работы с аппаратной частью ПК. Используется в повседневной работе разработчиками Gentoo.

Все дистрибутивы представлены ISO-образами, который можно записать на отдельный носитель, и загрузить в live-режиме прямо с LXF DVD. У вас присутствует возможность установить на жесткий диск.

КОММЕНТАРИЙ Прислайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДВОЕКЛЕТНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данной диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым другим 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, представляющего собой программу или данные. Прежде чем устанавливать какие-либо ПО на компьютер, пожалуйста, скачайте программу для загрузки с ее официального сайта.
Тираж издательства ООО «Марком», 188652, Россия, Ленинградская область, Всеволожский р-н, дер. Юрки, Школьная ул., 7-а. Лицензия ИДТР ВАР N 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 перенесут 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



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № Ф077-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж печатной версии 1000 экз., распространение электронной версии 30000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ**Главный редактор**Кирилл Степанов info@linuxformat.ru**Литературный и выпускающий редактор**

Елена Толстякова

Переводчики

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 рекламеВладимир 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**Установщик Ubuntu** Джонни Бидвелл [Jonni Bidwell]jonni.bidwell@futurenet.com**Художественный редактор** Эфраин Эрнандес-Мендоса[Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com**Выпускающий редактор** Клифф 'Вечный' Хоуп [Cliff 'Eternal' Hope]clifford.hope@futurenet.com**Директор редакции** Грэм Барлоу [Graham Barlow]**Старший художественный редактор** Джо Гулливер [Jo Gulliver]**ПОДГОТОВКА МАТЕРИАЛОВ**

Матс-Таге Аксельссон [Mats Tage Axelsson], Пол Бич [Paul Beech], Джонни Бидвелл [Jonni Bidwell], Нейл Ботвик [Neil Bothwick], Алекс Кокс [Alex Cox], Алан Декстер [Alan Dexter], Кент Ельчук [Kent Elchuk], Джереми Лэйд [Jeremy Laird], Уилл Мейстер [Will Meister], Брайан Морк [Brian Mork], Бобби Мосс [Bobby Moss], Ник Пирс [Nick Peers], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Валентин Синицын [Valentine Sinityn], Александр Толстой [Alexander Tolstoy], Михалис Цукалос [Mihalis Tsoukalos], Филиппа Уорр [Philippa War], Евгений Балдин, Евгений Гольшев, Андрей Гондаренков, Дмитрий Кузнецов, Алексей Федорчук

Иллюстрации Шейн Коллиндж [Shane Collinge]**Иллюстрация с обложки** magictorch.com**КОНТАКТНАЯ ИНФОРМАЦИЯ****UK:** Linux Format, 30 Monmouth Street, Bath BA1 2BWТел. +44 (0) 344 848 2852, email: linuxformat@myfavouritemagazines.co.uk**РОССИЯ:** Санкт-Петербург, пр. Медиков, 5, корп. 7Тел. +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 тек или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвалдса [Linus Torvalds]. "GNU/Linux" заменяется на "Linux" в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журнале, издаваемых Future plc group company, обращайтесь на сайт www.futureplc.com.

LXF238
Читайте
в июльском
номере



Как самим собрать... Рабочий стол

Ubuntu Gnome уже успел надоест?
Ну так соберите себе новый рабочий стол,
следуя нашему полному руководству!

WINE — не эмулятор!

А что же тогда?.. Тестируем функции свежей v3.x и показываем,
как выжать из них максимум.

Google TensorFlow

Прибавим ума своему Raspberry Pi, запустив на нем TensorFlow
для машинного обучения!

Улучшим WordPress

Вы сделали сайт, клонировали его и снабдили темой.
Теперь пора разогнать его до бешеной скорости.

Редакторы изображений

Под микроскоп *Сравнения* попали пять лучших редакторов
изображений, для детального изучения.

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

SAAYHEI

НОВОМУ ДАТА-ЦЕНТРУ
HETZNER ONLINE
В ХЕЛЬСИНКИ

Выделенный сервер
PX61-NVMe

- ✓ Intel® Xeon® E3-1275 v5 Quad-Core Skylake
 - ✓ Incl. Hyper-Threading-Technology
 - ✓ 64 ГБ DDR4 ECC RAM
 - ✓ 2 x 512 ГБ NVMe Gen3 x4 SSD
 - ✓ 100 ГБ место для резервных копий
 - ✓ 30 ТБ трафик*
 - ✓ местоположение Финляндия
- Установка 10.540 рублей



Выбери **ФИНЛЯндию**
для твоего сервера и сэкономь

Благодаря низким ценам на электроэнергию и прохладному климату, это новое местоположение дата-центра означает, что вы получите еще более низкие цены с такой же качественной инфраструктурой и обслуживанием. Кроме того, новое расположение даст вам еще лучшее покрытие сети и географическую избыточность для ваших систем, а также оно уже соответствует требованиям новых правил по защите данных Европейского Союза.

4.730 рублей в месяц

ru.hetzner.com

* Нет платы за превышение. При превышении 30 ТБ/месяц скорость соединения ограничивается (подсчёт ведётся по исходящему трафику, входящий и внутренний трафик не учитывается). Опционально можно снять ограничение, подтвердив оплату 89 руб. за каждый дополнительный ТБ.

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.

теперь в Финляндии