

DVD ДАРОМ!



ИЗУЧИТЕ DEBIAN 9.0

# LINUX FORMAT

Главное в мире Linux

Сентябрь 2017 №9 (227)

## ВИРТУАЛЬНО В LINUX!

Запускайте всё на одном ПК

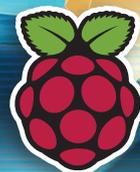
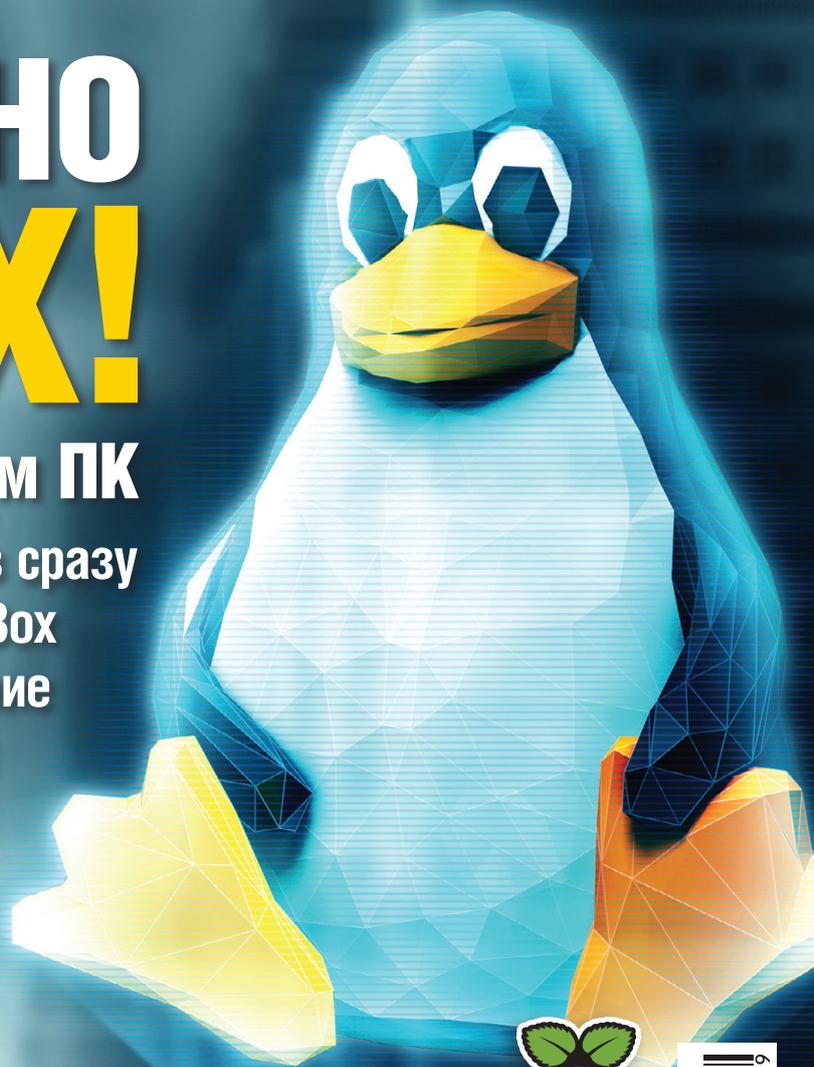
- » Приютите много дистрибутивов сразу
- » Извлеките максимум из VirtualBox
- » Улучшите графическое ускорение
- » Примените Docker, QEmu и KVM

**67** страниц учебников и статей!

- » Перевод Ubuntu на ZFS
- » Биткойны и блокчейны
- » BSD против Linux: кто кого?



Академия кодига: Учимся копировать на Python



### Жизнь на краю

« Традиционная раздробленность сферы встроенных технологий может стать препятствием для компаний »

Canonical — о том, как победить в войне IoT

### Плюс: Для Pi

7 страниц в копилку Raspberry Pi

- » Разработку скрасит Thonny
- » Следим за Twitter с Node-RED
- » Строим wiki-сервер на Pi Zero W



Шифрование

#### Защита файлов

- » Шифрование LUKS мигом скроет все ваши файлы



Сравнение

#### Бизнес-серверы

- » Какому дистрибутиву доверить свой малый бизнес?



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

Linux center  
www.linuxcenter.ru



# iTeleRadio

ИНТЕРАКТИВНОЕ РАДИО&ТЕЛЕВИДЕНИЕ



## Что мы делаем

- » Мы поддерживаем открытое сообщество, предоставляя источник информации и площадку для обмена мнениями.
- » Мы помогаем всем читателям получить от Linux максимум пользы, публикуя статьи в разделе «Учебники»: здесь каждый найдет что-то по своему вкусу.
- » Мы выпускаем весь код, появляющийся на страницах раздела «Учебники», по лицензии GNU GPLv3.
- » Мы стремимся предоставлять точные, актуальные и непредвзятые сведения обо всем, что касается Linux и свободного ПО.

## Кто мы

На сей раз мы спросили наших экспертов: в этом номере мы рассматриваем виртуализацию ВСЕГО! А какую сферу жизни вы сами хотели бы виртуализовать?



### Джонни Бидвелл

А нельзя ли виртуализовать наше правительство? И вносить наши личные изменения, и по моментальным снимкам определять, что является хорошим решением, а что нет, и соответственно свергать виновных. Думаю, нам не позволили бы этого сделать, но наблюдать за протестами виртуализованных лидеров в парламентских дебатах стало бы интересно.



### Ник Пирс

Я делаю далеко не всё, чтобы поддерживать форму, и, наверное, виртуализация упражнений помогла бы мне оставаться относительно подтянутым без требуемой для этого тяжелой работы. Ну да ладно, я средой не выглядел хорошо в шортах! Кроме того, я мог бы улучшить свою футбольную статистику, став хоть вполноценно конкурентоспособным.



### Лес Паундер

Очень бы пригодились голодеки, как в «Звездном пути». В конце концов, было бы неплохо путешествовать по галактике... постоянно путешествуя по галактике. Интересно, работает ли Linux на Enterprise? Было бы очень жаль, если бы во время битвы произошла паника ядра, или зависимости пакетов бронезащиты не были удовлетворены.



### Найт Дрейк

Когда пару лет назад я зашел в офис агентов по недвижимости, мне выдали гарнитуру VR для изучения виртуальных версий различных объектов на продаже. Я мигом сообразил, что мне в общем-то незачем переезжать. Вместо этого надо просто взять гарнитуру и превратить мою односпальную квартирку в славное виртуальное историческое здание!



### Маянк Шарма

Было бы здорово, если бы я мог занести денежный аспект моей жизни в контейнер и вызывать скрипт, чтобы каждый час порождать несколько экземпляров. И я хотел бы работать в безголовом режиме, без атак отвлекающих мыслей, особенно про надвигающийся срок сдачи статей, который маячит в углу комнаты.



### Валентин Синицын

Начинать всегда следует с себя. Один я отправился на работу, второй я остался в это время дома с семьей, третий я отвечает на этот вопрос, четвертый я веселился в баре с друзьями. А я, который на самом деле я, спокойно лежит себе на диване и отсыпается после праведных трудов.



## Забывтый юбилей

» Недавно я с удивлением обнаружил, что не могу вспомнить, когда последний раз видел серверное приложение, выполняемое на «физическом железе». Не только в корпоративных ЦОД, но даже в самых мелких компаниях, где мне приходилось бывать, гипервизоры полностью вытеснили с серверов «классические» ОС.

Интересная история самого термина «гипервизор», у которого в этом году юбилей. 50 лет назад IBM выпустила на рынок операционную систему CP-67, позволяющую выполнять несколько независимых экземпляров ОС на одном компьютере System/360-67. Дальний потомок этой системы жив и называется z/VM. Мне даже пару раз приходилось ставить Linux на его виртуальные машины. В операционках IBM ядро еще с начала 1960-х принято было называть «супервизор» — ну, а то, что управляет несколькими супервизорами... простая игра слов.

15 лет назад основатели компании VMware перенесли эту идею на архитектуру x86, выпустив первые версии «серверных» гипервизоров ESX и GSX. Потом вышли и другие разработки, проприетарные и свободные. Виртуализация серверов сразу решила две первоочередных проблемы: уменьшение числа физических серверов, что снижает расходы на электропитание и кондиционирование ЦОД, и упрощение управления ИТ-инфраструктурой. Администрирование служб и приложений на «настоящем железе» уже вспоминается как страшный сон.

Сегодня мало кто помнит о CP-67, а вот термин «гипервизор» вошел во все языки без перевода.

### Кирилл Степанов

Главный редактор

» [info@linuxformat.ru](mailto:info@linuxformat.ru)

## Как с нами связаться

Письма для публикации: [letters@linuxformat.ru](mailto:letters@linuxformat.ru)

Подписка и предыдущие номера: [subscribe@linuxformat.ru](mailto:subscribe@linuxformat.ru)

Техническая поддержка: [answers@linuxformat.ru](mailto:answers@linuxformat.ru)

Общие вопросы: [info@linuxformat.ru](mailto:info@linuxformat.ru)

Проблемы с дисками: [disks@linuxformat.ru](mailto:disks@linuxformat.ru)

Вопросы распространения: [sales@linuxformat.ru](mailto:sales@linuxformat.ru)

Сайт: [www.linuxformat.ru](http://www.linuxformat.ru), группа «ВКонтакте»: [vk.com/linuxform](https://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](http://linuxformat.ru/avtoram.phtml)

Зарегистрируйтесь  
на сайте  
[shop.linuxformat.ru](http://shop.linuxformat.ru)



СОМНЕНИЯ

Скачайте бесплатно  
архивные PDF-номера  
журнала с сайта  
[linuxformat.ru/archive](http://linuxformat.ru/archive)  
и загляните на  
[wiki.linuxformat.ru](http://wiki.linuxformat.ru)



Версия для iPad  
и iPhone доступна  
в App Store

Выберите вид подписки

PDF-версия  
на 6 месяцев

990 ₺

PDF-версия  
на 12 месяцев

1800 ₺

Печатная версия  
на 6 месяцев

1890 ₺

Печатная версия  
на 12 месяцев

3480 ₺

Выберите вид доставки

Оплатите

# Читайте Linux Format!

Станьте Linux-гуру

## PDF-версия журнала Linux Format подойдет для тех, кто:

- Заботится о соблюдении прав деревьев
- Любит читать с экрана
- Мечтает получать каждый номер в день выхода журнала
- Хочет бесплатно скачивать содержимое DVD-приложения к каждому номеру

## Печатная версия Linux Format понравится читателям, которые:

- Любят читать бумажные журналы
- Хотят получить в подарок подписку на PDF-версию Linux Format...
- ...а также диск с архивом журнала 2005–2014 гг.
- Порадуются новинкам открытого ПО на DVD-приложении к Linux Format в каждом номере

## Способы доставки

- Курьером «ГНУ/Линуксцентра» по Москве и Петербургу
- Курьерской службой СПСР по России
- Почтой по России заказной или простой бандеролью
- Самовывоз из офиса «ГНУ/Линуксцентра» в Санкт-Петербурге
- Через пункты выдачи интернет-магазинов [iml.ru](http://iml.ru) в 11 городах России: Санкт-Петербург, Москва, Екатеринбург, Калуга, Нижний Новгород, Орел, Ростов-на-Дону, Тверь, Тюмень, Челябинск, Ярославль

## Способы оплаты

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



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

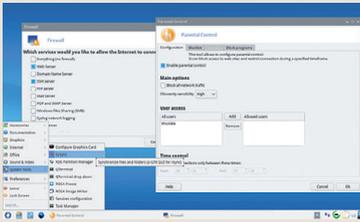
# Содержание

«Две вещи бесконечны: Вселенная и людская глупость; но насчет Вселенной я не уверен.» Альберт Эйнштейн

## Обзоры

### Rosa R9 LXQt ..... 14

Побаловать нас нетрудно: ищем всего лишь дистрибутив, который глубоко настраивается и является стабильным, да еще отличается продуманными графическими приложениями для администратора системы.



➤ Администраторы — администраторами, но за детками тоже нужен пригляд.

### Tanglu 4.0 Gnome ..... 15

И кто только удумывает подобные имена? Тем не менее, мы взялись устроить тест-драйв этому дистрибутиву на базе Debian. Ну ни рыба ни мясо... уж лучше останемся при своих.

### Robolinux 8 Cinnamon ..... 16

Дистрибутив был учрежден с гуманнейшей целью: помочь беженцам с Windows. И вдруг скатился к корыстным мотивам. Неужели это заразно?

### AMD Ryzen 5 1400 ..... 17

Дешевый четырехъядерный процессор от AMD замахнулся на конкуренцию с Intel. Выбор неплохой, хотя при скромном бюджете стоит подождать Ryzen 3.

### Dawn of War III ..... 18

В сорокатысячной вселенной Warhammer есть, по сути, только война — даже перекусить времени не выкроишь. Космические орки и Император Человечества явно расставили приоритеты неправильно.



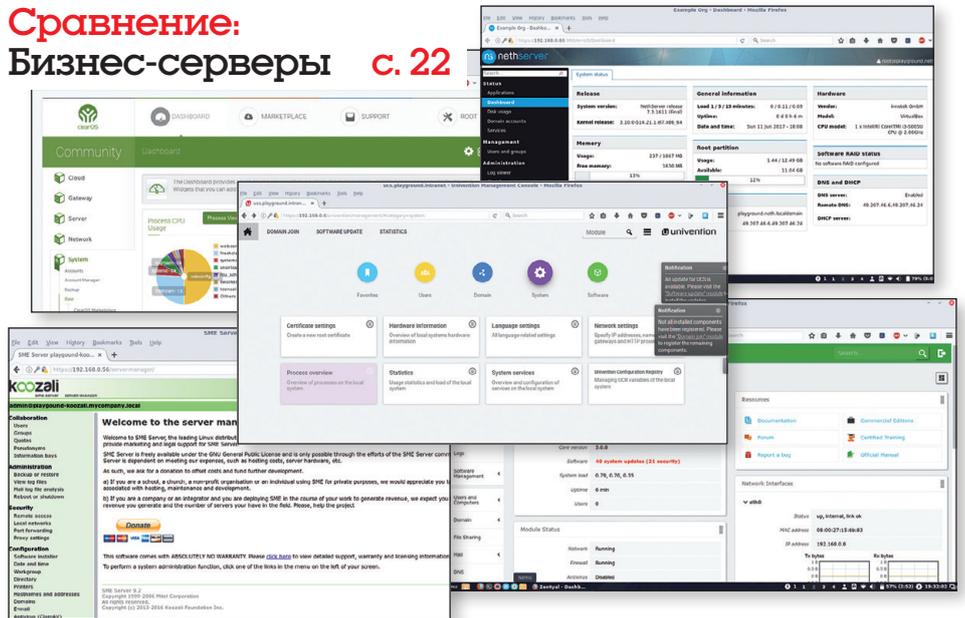
➤ Какой прок от войны? Прибыли корпораций и раскрутка политиков.

## ВИРТУАЛИЗУЕМ LINUX!

Тукс внутри Тукса внутри Тукса... идем в странный мир виртуальности, где системы запускаются внутри систем. Объясним, как быстрее, проще и лучше виртуализоваться, на с. 28



## Сравнение: Бизнес-серверы с. 22



## Интервью



«Настольная версия продолжает быть ключевой для бизнеса и сообщества»

Тибо Руффино — про устремления Canonical с. 38

# На вашем бесплатном DVD

**Debian 9.0**  
Исследуйте прочный как скала дистрибутив серверного класса для всех!  
Новый релиз

**Voyager Live 9.0**  
Дружелюбный дистрибутив Linux с предустановленным рабочим столом  
Новый релиз

**LIVE-ДИСК: ГОТОВ К РАБОТЕ**  
ВСЕ НЕОБХОДИМОЕ ДЛЯ СТАРТА В LINUX

## Debian 9.0 Gnome 64-битный Voyager Live 9.0 64-битный

### Бизнес-дистрибутивы

» Каждый месяц — только лучшие дистрибутивы

ПЛЮС: HotPicks, код и библиотека с. 106

Побалуйте себя и любимых подпиской на LXF!



Доступно в AppStore!



www.linuxformat.ru/subscribe

## Пользователям Raspberry Pi



- Новости Pi** ..... 92  
Ubuntu Core повязали с Pi 3 Compute Module, физики соорудили доску Гальтона на Pi, а Picorder принялся за кино съемки.
- Thonny** ..... 93  
IDLE — не один в поле воин; у этой среды разработки на Python нашлась альтернатива.
- Тревога по хэштегу** ..... 94  
Мониторим Твиттер и создаем систему оповещения на языке Node-RED.
- Wiki-сервер на Pi Zero W** ..... 96  
Беспроводная википедия у вас в кармане! Причем по карману не ударит.



## Ищите в номере

- BSD против Linux** ..... 42  
Кто кого соборет? Пристально и беспристрастно проанализируем оба ядра с открытым кодом; но Linux-то ведь всё равно лучше, правда?
- Лучший ноутбук с Linux** ..... 46  
Разберемся, как такой купить, как установить на нем Linux самому и как собрать ноутбук исключительно из открытого оборудования, на радость Ричарду Столлмену и, конечно, себе!
- Сайты на Tor** ..... 50  
Строим безопасные сайты в таинственном полумраке Dark Web, пользуясь луковичной маршрутизацией. Запускаем скрытый сервис — ну, чтоб никто не догадался...

## Академия кодига

- Загадка LXF!** ..... 84  
Мы приняли число π близко к сердцу — и вот затеяли его расчет с точностью до произвольного количества десятичных знаков, сразу на нескольких языках программирования.
- сору на Python** ..... 88  
К концу нашего классического упражнения по созданию аналога утилиты копирования cp на Python вы волей-неволей заделаетесь системным программистом. Итак, повышайте свою квалификацию!

```
18 getcontext().prec=numberOfDigits
19
20 def Pi(n):
21     t = Decimal(0)
22     pi = Decimal(0)
23     deno = Decimal(0)
24     k = 0
25     n = int(n / 20)
26     for k in range(n):
27         t = ((-1)**k)*(factorial(6*k))*((13591409+545140134*k)
28         denominator = factorial(3*k)*(factorial(k)**3)*(640320**(3*k))
29         pi += Decimal(t)/Decimal(denominator)
30         if printAll == True:
31             print(pi)
32     pi = pi + Decimal(12)/Decimal(640320**Decimal(1.5))
33     pi = 1/pi
34     return pi
35
36 print(Pi(numberOfDigits))
```

## Учебники

- Основы терминала**
- Линуксоиды шутят** ..... 56  
В жизни есть место не только серьезности, и даже терминал можно превратить в забаву!
- Утилиты Linux**
- Надзор за процессами** ..... 58  
Рассматриваем программы, позволяющие наблюдать за процессами и управлять ими.
- Поддержка ФС в Linux**
- Что такое VFS** ..... 62  
Файловых систем в Linux — несть. Но все они накрываются одним уровнем абстракции...
- Восстановление**
- Лечит Rescatux** ..... 64  
Исцеляем захворавшие операционные системы! С вами — доктор Тукс.
- Шифрование**
- Защита LUKS** ..... 68  
Эта система способна обезопасить ваши файлы и устройства. И уже встроена в Linux.
- Биткойн**
- Полезьа блокчейнов** ..... 72  
С помощью блокчейнов ставим отметки времени, выпускаем акции и эмитируем валюту.
- Файловые системы**
- Переходим на ZFS** ..... 76  
Эта файловая система справляется с зеттабайтами данных. Но ей требуется приличное оборудование.
- Интернет вещей**
- Стать Большим Братом** ..... 80  
Визуализируем процессы, происходящие в Послушном Доме, контроля ради.

## Постоянные рубрики

- Новости** ..... 6  
WD представила RAID-накопитель на 20 ТБ, Solar inRights признана ФСТЭК, пакет NVIDIA Quadro vDWS творит чудеса, в МФТИ исследуют искусственный интеллект, HPE и NASA запустили суперкомпьютер в космос, растет спрос на специалистов по СПО, Dell обновила Inspiron, а РОСА укрепляет обороноспособность.
- Вести мобильных ОС** .... 20  
Verizon, Ericsson и Qualcomm осилили скорость связи 1 Гбит/с, Intel реализует машинное зрение и вместе с Mobileye создает тестовый автопарк, а ноутбуки от Intel ускорятся чуть ли не вдвое.
- Сравнение** ..... 22  
Как бы учителя действительно не сбежали в бизнес, при таких-то серверах: ClearOS, Kozaali SME Server, NethServer, Univenton Corporate Server, Zentyal.
- Интервью LXF** ..... 38  
Майк Белл и Тибо Руффино делятся информацией о последних новшествах Canonical. Нет, настольный Ubuntu никто ликвидировать не собирается!
- HotPicks** ..... 100  
Отведаете горяченького! Лучшие в мире новинки свободного ПО: *Butterfly Effect, Frogatto, G'MIC-Qt, Gsmartcontrol, HTTraQt, Krita, KWave, Ring, Scribus, SmartDeblur, Ubunsys.*
- Диск Linux Format** ..... 106  
Содержимое двустороннего DVD этого месяца.
- Пропустили номер? ...** 108  
Душит вредоносное ПО? Надоела слежка? Необходимы обновления? Срочно бежим за LXF226, и долой Windows 10!
- Через месяц** ..... 112  
Снова в школе! Вооружаем вас открытым инструментарием — и программным, и аппаратным — для выполнения учебных планов.



**В ЭТОМ НОМЕРЕ:** Объемы от WD » IGA с сертификатом » Серверы ликуют » ИИ-суперкомпьютер » Компьютер на Марсе » СПО в тренде » Трансформер от Dell » ПО для вояк

## НАКОПИТЕЛИ

# RAID-массив на столе

WD представила внешний накопитель на 20 ТБ.

**К**омпания Western Digital представила MyBook Duo — настольный RAID-накопитель, идеально подходящий для работы с большими объемами данных и представляющий интерес как для домашних пользователей, так и для малого бизнеса. MyBook Duo состоит из двух жестких дисков WD Red, оптимизированных для работы в RAID и обеспечивающих скорость последовательного чтения 360 МБ/с. Устройство оснащено встроенным ПО для работы с RAID-массивами, комплектуется утилитами WD Discovery для резервного копирования и WD Security для защиты данных, поддерживает аппаратное 256-битное AES-шифрование, оснащено портом USB-C 3.1 gen2 и двумя портами USB-A для подключения дополнительных аксессуаров.

MyBook Duo поставляется с трехлетней ограниченной гарантией в конфигурациях 4 ТБ (\$259), 6 ТБ (\$289), 8 ТБ (\$329), 12 ТБ (\$419), 16 ТБ (\$599) и 20 ТБ (\$799).

«И дома, и в офисах, современные пользователи создают и сохраняют огромные объемы данных. Необходимы высокопроизводительные и высокоскоростные решения, позволяющие легко и надежно перемещать и хранить эти данные в одном месте, — считает Свен Ратджен [Sven Rathjen], вице-президент по маркетингу Client Solution (подразделение Western Digital Corporation). — Мы наблюдаем этот процесс в новых сегментах создания контента: рынок БПЛА, фотографии и видеоролики, производящие и редактирующие высококачественные фото- и видеофайлы. Простое

» **My Book Duo, настольный накопитель от WD, предназначен для быстрого и надежного сохранения больших объемов данных в условиях дома или офиса.**

в использовании, высокочемое настольное решение для хранения данных, каким является My Book Duo, позволяет создателям подобного контента идти в ногу с их растущими требованиями к накопителям».



» Рубрику готовил  
**АНДРЕЙ  
ГОНДАРЕНКОВ**

## ОТЕЧЕСТВЕННОЕ ПО

# Solar inRights: защита на всех уровнях

Российская IGA-платформа сертифицирована ФСТЭК РФ.

**S**olar Security, разработчик продуктов и сервисов для целевого мониторинга и оперативного управления информационной безопасностью, сообщает, что решение класса IGA (Identity Governance and Administration) Solar inRights получило сертификат ФСТЭК России. Сертификат соответствия №3793 подтверждает, что решение Solar inRights 2.0 отвечает требованиям ФСТЭК России по 4-му уровню контроля и технических условий. Полученный

сертификат позволяет применять Solar inRights при создании автоматизированных систем до класса защищенности 1Г включительно, а также для защиты информации в информационных системах персональных данных (ИСПДн) всех уровней защищенности. Solar InRights входит в Единый реестр российского ПО для электронных вычислительных машин и баз данных.

Solar inRights обеспечивает автоматическое исполнение регламентов управления

доступом, а также управление полным жизненным циклом учетных записей, ролей, информационных систем и других субъектов управления. Решение позволяет снизить риски, связанные с ошибками исполнения процедур предоставления доступа, избыточными правами сотрудников, а также повышает прозрачность процессов управления доступом.

«Solar inRights — это российский продукт с уровнем зрелости западного решения класса IGA. Теперь организации, которые используют исключительно сертифицированные решения, могут использовать передовые технологии по управлению правами доступа, автоматизировать ряд рутинных операций службы ИТ и ИБ, и главное — снизить риски, связанные с избыточными правами доступа сотрудников к корпоративным информационным системам», говорит Дмитрий Бондарь, руководитель направления Solar inRights компании Solar Security.

НОВОСТИ NVIDIA

# Для ИИ и VR

Пакет Quadro vDWS превращает серверы с GPU архитектуры Pascal в рабочие станции.

**N**vidia представила программный пакет NVIDIA Quadro Virtual Data Center Workstation (Quadro vDWS), позволяющий превратить сервер с GPU Tesla в мощную рабочую станцию. Quadro vDWS запускает приложения в виртуализированной среде и унифицирует графические и вычислительные процессы, что значительно повышает производительность при работе с проектами 3D, фотореалистичного рендеринга, виртуальной реальности и глубокого обучения.

Quadro vDWS работает с GPU архитектуры Pascal и позволяет:

- » повысить графическую производительность по сравнению с предыдущей архитектурой GPU NVIDIA на 50%;
- » задействовать рабочие процессы CUDA и Open CL для оптимизации моделирования и автоматизированного проектирования;
- » создавать сложные и фотореалистичные проекты комплексного трехмерного

моделирования, предоставляя до 24 ГБ памяти GPU для больших моделей;

- » повысить производительность для пользователей Linux за счет перемещения ресурсоемкого кодирования H.264 в облако.

«Благодаря Quadro vDWS, установленному на серверах с GPU Tesla, предприятия смогут работать с еще большими массивами данных, выполнять самые требовательные приложения и повысить общую мобильность работы», уверяет вице-президент подразделения Professional Visualization NVIDIA Боб Петт [Bob Pette]. Решение уже доступно на 100 серверах по всему миру, включая предложения от Cisco, Citrix, Dell, HP, Lenovo и VMWare.

NVIDIA усовершенствовала свою технологию виртуальных десктопов NVIDIA GRID virtual PC (vPC), так что теперь:

- » GPU Tesla P40 поддерживает до 24 экземпляров 1-ГБ профилей виртуальных рабочих столов (в 2 раза больше, чем Tesla M60);

» Пакет NVIDIA Quadro vDWS обеспечивает виртуализированную работу с графическими и вычислительными задачами за счет ресурсов серверов с GPU Tesla.



- » возможен мониторинг vGPU на всех уровнях от хоста до конечной точки, выполнена интеграция с VMware vRealize Operations (bROps), Citrix Director и XenCenter;

- » есть новый планировщик ресурсов GPU.

А в линейку GPU Tesla P4, P40 и P100 NVIDIA добавила новый продукт Tesla P6, который предназначен для blade-серверов и предлагает 16 ГБ памяти (возможна реализация до 16 экземпляров 1-ГБ профилей виртуальных десктопов).

ВНЕДРЕНИЯ

# Первый в мире

В МФТИ установили суперкомпьютер для исследований в области ИИ.

**В** Лаборатории нейронных систем и глубокого обучения МФТИ появился первый в мире суперкомпьютер, спроектированный специально для обучения искусственных нейронных сетей. В основе суперкомпьютера DGX-1 от NVIDIA лежит новое поколение GPU, которые обеспечивают скорость обработки данных в задачах ИИ, сравнимую с 250 серверами архитектуры x86. Он оснащен всем необходимым аппаратным и программным обеспечением для задач глубокого обучения и набором инструментов разработки, и совместим с популярными аналитическими приложениями с поддержкой GPU.

«Чем более мощное „железо“ есть в нашем распоряжении, тем с более сложными нейросетевыми архитектурами мы сможем работать. Сложность модели зачастую позволяет совершить революционный скачок в решении практических задач. Так, например, текущая революция в компьютерном зрении и распознавании речи связана, в том

числе, с ростом вычислительных возможностей. Хорошее оборудование позволит решать практические задачи, за которые без него мы бы даже взяться не смогли», рассказывает заведующий Лабораторией нейронных систем и глубокого обучения МФТИ Михаил Бурцев.

Мощности данного суперкомпьютера будут, в основном, задействованы в работе над проектом по созданию разговорного искусственного интеллекта iPavlov, который лаборатория реализует в рамках Национальной технологической инициативы совместно со Сбербанком. Ученые Физтеха занимаются разработкой «разговорного» машинного интеллекта, который будет способен вести содержательный диалог с человеком. Алгоритм сможет не только отвечать на вопросы собеседника, но и запрашивать информацию, необходимую для того, чтобы решить поставленную в диалоге цель. Для этого нейронная сеть будет проходить свое «обучение» на больших



» Последние моменты установки вычислительного узла DGX-1 в серверную стойку.

массивах документов и текстовых записей диалогов между людьми. Помимо лаборатории Бурцева, на суперкомпьютерах смогут работать и другие научные коллективы института, которые также исследуют искусственные нейронные сети для разработки новых лекарств.

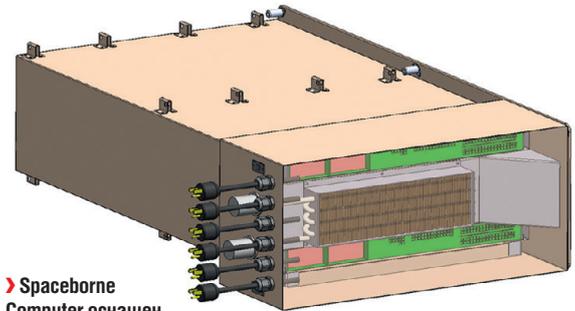
ЭКСПЕРИМЕНТЫ

# Через тернии — к Марсу!

HPE и NASA будут тестировать на МКС «космический» суперкомпьютер.

Из Космического центра Кеннеди во Флориде 14 августа стартовала ракета SpaceX CRS-12 компании Илона Маска, доставившая на МКС корабль SpaceX Dragon с суперкомпьютером Spaceborne Computer производства Hewlett Packard Enterprise на борту. Spaceborne («Космический») выполнен на системах класса HPE Apollo 40 с высокоскоростной коммутационной сетью и работает под управлением ОС Linux; он будет установлен в Национальной лаборатории МКС. Целью эксперимента HPE и NASA является проверка возможности безотказной работы Spaceborne Computer в космосе в течение года — примерная длительность марсианской миссии. Из-за ограничений на компьютерные вычисления в космосе, сегодня многие из них выполняются на Земле. Однако при полете к Марсу задержка между отправкой сигнала на Землю и откликом на него — около

40 минут. «Коммуникационный сдвиг делает управление техникой с Земли сложным и потенциально опасным в критически важной ситуации, которую астронавты не смогут разрешить сами, — пишет в своем блоге старший вице-президент HPE Ален Андреоли [Alain Andreoli]. — Миссия потребует мощных бортовых вычислительных ресурсов с долгим сроком безотказной работы. Несомненно, результаты эксперимента улучшат высокопроизводительные вычисления (HPC) на Земле и повлияют на другие области технологических инноваций». Коммерчески доступная высокомоощная компьютерная система отправилась в космос впервые. По требованиям NASA, космические компьютеры должны обладать повышенной устойчивостью к воздействию неблагоприятных условий: радиации, солнечных вспышек, микрометеоритов, субатомных частиц, нестабильности питания



Spaceborne Computer оснащен специальным системным ПО, нейтрализующим угрозы внешней среды и обеспечивающим повышенную надежность вычислений.

и охлаждения на борту. Вместо дорогих и громоздких модификаций аппаратных средств, HPE защитила свой суперкомпьютер системным ПО, управляющим отладкой его узлов в реальном времени, а также сглаживает последствия ошибок, вызванных внешними условиями. Суперкомпьютер заключен в уникальный контейнер с водяным охлаждением.

НЕМНОГО СТАТИСТИКИ

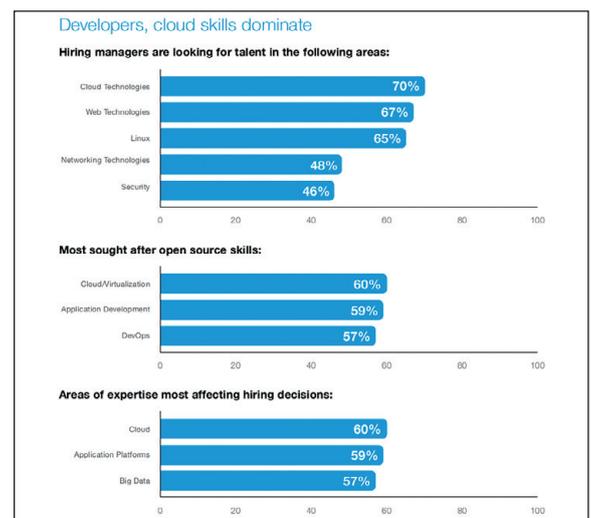
# Я б в IT-шники пошел!..

Спрос на квалифицированных специалистов по открытому ПО неуклонно растет.

Согласно ежегодному отчету Open Source Jobs Report 2017, опубликованному консорциумом The Linux Foundation и сайтом Dice.com, для менеджеров по найму персонала поиск специалистов по Open Source становится всё более приоритетным направлением деятельности; 2/3 менеджеров заявили, что штат сотрудников в открытых проектах будет увеличиваться значительно сильнее, чем в других сферах деятельности их компаний. В качестве основных причин указываются расширение бизнеса (60%), увеличение количества используемых открытых технологий (42%), переход на открытое ПО как основу бизнеса (30%). В свою очередь, 86% IT-специалистов считают, что именно благодаря знанию открытых продуктов они смогли сделать карьеру; 52% не сомневаются, что смогут легко найти другую работу; только 27% за последнее полгода не получили ни одного предложения от рекрутинга. Облачные и виртуализационные технологии, разработка приложений и web-разработка, Linux и Big

Data — профессионалы в этих областях признаны наиболее востребованными; далее расположились организация и эксплуатация сетей и информационная безопасность. Самыми востребованными специалистами являются разработчики (73%), инженеры DevOps (60%) и системные администраторы (53%). По-прежнему высок спрос на специалистов по Linux, поскольку всё чаще в качестве основной платформы бизнес использует open-source системы (например, наиболее популярные инструменты DevOps — Chef, Puppet и Ansible; то же самое можно сказать об Amazon Web Services и Google Cloud Platform; на платформе Microsoft Azure 30% виртуальных машин работают под управлением Linux, и эта цифра стремительно растет). Специалистам по Open Source всё чаще приходится иметь дело и с контейнерными технологиями виртуализации.

64% менеджеров отметили, что их сотрудники активно повышают квалификацию на различных курсах, тренингах и т. п. (по сравнению с 55% в 2016 г.);



Специалисты в облачных технологиях и виртуализации, разработчики приложений и инженеры DevOps представляют всё больший интерес для менеджеров по найму персонала.

увеличивается доля компаний, готовых оплачивать сотрудникам процесс получения сертификации.

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](http://unixedu.ru)

**UnixEducationCenter**

МОДЕРНИЗАЦИЯ ОБОРУДОВАНИЯ

# DELLaet Dell

В октябре начнутся продажи обновленной линейки ноутбуков Inspiron с процессорами Intel Kaby Lake Refresh.

Компания Dell стала одним из первых производителей, оснастивших свои ноутбуки процессорами Intel следующего поколения: уже в сентябре в продажу поступила обновленная модель хорошо зарекомендовавшего себя ноутбука Dell XPS 13 с CPU Core 8-го поколения, в котором удвоено количество процессорных ядер; сам ноутбук стал более тонким и легким, но и начальная цена у него выросла до \$1399,99 (наименее дорогостоящая конфигурация XPS 13 с процессором Core 7-го поколения предлагается за \$799).

В семействе ноутбуков Inspiron 5000 и 7000 серий обновлены гибридные и стандартные модели; все они получили процессоры Intel Core 8-го поколения. 15-дюймовая модель трансформера Inspiron 7000 опционально может комплектоваться накопителем PCIe NVMe SSD, 4K-дисплеем

3840 × 2160, дискретной видеокартой Nvidia GeForce 940MX с 2 или 4 ГБ памяти. 17-дюймовая модель Inspiron 7000 2-в-1 комплектуется дисплеем Full HD, процессором Intel Core i5 или i7, видеокартой NVIDIA GeForce MX150 с 2 ГБ видеопамяти GDDR5, до 16 ГБ RAM, накопителем HDD или SSD, аккумулятором с автономностью до 11 часов, портами USB Type-C и USB 2.0, карт-ридером microSD. Стартовые цены моделей в линейке Inspiron 7000 составят от \$799 до \$949.

Стартовая цена 13- и 15-дюймовых трансформеров Inspiron 5000 с процессорами 8-го поколения начнется с \$799,99 за модель с активным стилусом и \$749,99 без него. В отличие от старшей линейки, дисплеи здесь только Full HD, установка дискретной видеокарты не предусмотрена.

Все обновленные модели серии Inspiron поступят в продажу в октябре.

СОБЫТИЯ

# И для созидания, и для обороны

На МВТФ «АРМИЯ-2017» «НТЦ ИТ РОСА» представила арсенал российского ПО для информационной безопасности бизнеса и государства.

На международном военно-техническом форуме «Армия-2017», который проходил с 22 по 27 августа в подмосковной Кубинке, российский центр разработки решений на основе свободного программного обеспечения ООО «НТЦ ИТ РОСА» впервые представил всю линейку операционных систем, системных приложений и системы управления средой виртуализации ROSA общегражданского и специального назначения.

В линейке продуктов российской компании — настольные и серверные операционные системы с оригинальным эргономичным интерфейсом и полноценным набором существующего прикладного программного обеспечения. Также была представлена платформа виртуализации, позволяющая разворачивать виртуальный

Центр обработки данных — ROSA VIRTUALIZATION. Программное обеспечение разработки «НТЦ ИТ РОСА» предназначено, в том числе, и для обеспечения безопасности, защиты информации государственных и частных организаций.

Продуктовая линейка «НТЦ ИТ РОСА» включает операционные системы для личного, корпоративного и специального пользования. Компания также предлагает программные продукты, сертифицированные по требованиям регуляторов ФСТЭК и МО РФ. Сертифицированные решения «НТЦ ИТ РОСА» позволяют производить аттестационные испытания автоматизированных систем на обработку конфиденциальной информации, персональных данных и сведений, составляющих государственную тайну. **LXF**

Новости короткой строкой

16 августа скончался Андрей «ache» Чернов — автор кодировки KOI8-R, один из создателей Рунета и один из первых разработчиков FreeBSD. Источник: [techquisitor.dreamwidth.org](http://techquisitor.dreamwidth.org)

Тесты российского CPU Baikal-T1 (1,2 ГГц) показали: по ряду параметров его мощность сопоставима с аналогами от ведущих мировых брендов. Источник: [www.baikalelectronics.ru](http://www.baikalelectronics.ru)

МТС, МГТС и Nokia подтвердили возможности 10GPON как транспортной инфраструктуры для мобильных сетей 5G в московском регионе. Источник: [mgts.ru](http://mgts.ru)

В Android 8.0 Oreo реализована функция «картинка в картинке», переработано меню настроек, значительно сокращено энергопотребление. Источник: [www.blog.google](http://www.blog.google)

«Новые облачные технологии» и «НПО РусБИТех» совместно запустили дистрибутив Astra Linux Special Edition с пакетом «Мой офис». Источник: [www.cnews.ru](http://www.cnews.ru)

В августе доля настольного Linux впервые преодолела отметку в 3% и составила 3,37% (у Windows 90,7%, у MacOS 5,94%). Источник: [www.netmarketshare.com](http://www.netmarketshare.com)

Релиз ядра Linux 4.13 поддерживает до 2 млрд файлов в папках ФС ext4; реализованы подсистема драйверов «mtd» и протокол TLS на уровне ядра. Источник: [lkm1.org](http://lkm1.org)

По новому экспортному соглашению Fedora Project, с 5.09.2017 Fedora не разрешен в Крыму. Источник: [russianfedora.org](http://russianfedora.org)

Основные изменения Service Pack 3 для SUSE Linux Enterprise 12 — поддержка нового оборудования, средств виртуализации и безопасности. Источник: [www.suse.com](http://www.suse.com)

Комитет ISO по стандартизации языка C++ утвердил спецификацию C++1z как международный стандарт C++17; она уже реализована в GCC и Clang, и частично в Intel C++ и Visual C++. Источник: [herbsutter.com](http://herbsutter.com)

У карты SanDisk Ultra microSDXC UHS-I от WD рекордная для такого формата емкость 400 ГБ; скорость передачи данных до 100 МБ/с, цена \$250. Источник: [www.theinqu](http://www.theinqu)

# 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](https://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](https://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](http://www.linuxcenter.ru)*



Новинки программного и аппаратного обеспечения в описании наших экспертов



**АЛЕКСЕЙ ФЕДОРЧУК**  
Тэг <сарказм>  
по умолчанию,  
смайлики по вкусу.

## Наконец-то о Maui

На протяжении последнего времени в этой колонке неоднократно упоминался дистрибутив Maui — и каждый раз с обещанием рассказать о нем подробнее вскоре. Вот и настало время исполнения обещаний. Так что Maui...

...это дистрибутив, основанный в базовой своей части на Ubuntu LTS текущего разлива (то есть в данный момент — на 16.04). Не использующий при этом технологию HWE — но обновляющий ядро по собственному графику, только в рамках LTS его версий. А в качестве десктопа применяющий KDE — да не простой, а Neop'ое последних версий (о котором тоже говорилось не раз). И опять же не «в лоб», а через промежуточный тестовый репозиторий. Чем обеспечивается гармония между актуальностью и устойчивостью. Что является первой особенностью дистрибутива. Вторая его особенность — отсутствие «заиклениости» на идеологической чистоте: несмотря на применение KDE как единственного десктопа, в штатном комплекте дистрибутива — множество GTK-приложений, органично интегрированных в среду Plasma. Есть у него и третья особенность: несмотря на внешнее сходство, он ни в коем случае не является Kubuntu. И, того паче, с последней несовместим, ибо использует гранаты... пардон, Qt-библиотеки иной системы. В общем, это хороший дистрибутив для всех, кто уважает современное KDE (Plasma от 5.8 и выше) и не страдает аллергией на Ubuntu. Но путем несложных действий его можно сделать еще лучше. Как — расскажем в следующей колонке. [alv@posix.ru](mailto:alv@posix.ru)

## Сегодня мы рассматриваем:

**Rosa R9 LXQt** ..... 14  
Настраиваемый дистрибутив с продуманными графическими приложениями для управления системой — более всего известен как «дистрибутив KDE», но на сей раз предпочел легковесный рабочий стол LXQt.

**Tanglu 4.0 Gnome** ..... 15  
Дистрибутив поддерживает драйверы устройств, проприетарные коды и множество встроенных программ. А если вы попытаетесь установить новое ПО, проблем не оберется. Лучше пока переждать с любым другим дистрибутивом.

**Robolinux 8.8.1** ..... 16  
Пытаясь адаптироваться к бежжанам с Windows, дистрибутив заралился

корпоративным корыстолюбием. Предлагать свободное ПО за деньг — ну где это видано? Хорошо хоть за техподдержку при установке не надо платить...

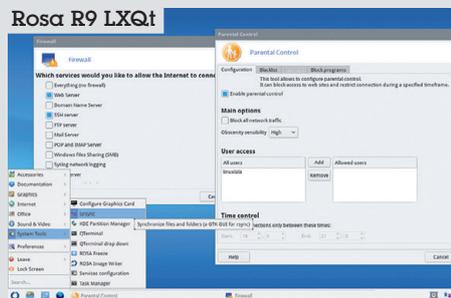
**AMD Ryzen 5 1400** ..... 17  
Новый процессор вполне оправдывает свои спецификации и выглядит довольно привлекательной покупкой, за такую-то цену. Но это если у вас уже есть материнская плата от AMD и приличная память.

**Dawn of War III** ..... 18  
Будь сюжет получше, игра, возможно, была бы просто великолепной, ведь все компоненты для этого есть. А так — при насыщенном действии скучать, конечно,

не придется, но необходимость неустанно проявлять агрессию утомляет.



➤ AMD Ryzen 5 1400 обеспечивает те показатели, которые вы в целом и ожидаете.



➤ Преимущество Rosa в том, что для обеспечения серьезной мощи требуется всего 512 МБ ОЗУ.



➤ Игра стремится во всем приблизиться к классике жанра стратегии реального времени.

## Сравнение: Бизнес-серверы с. 22

### ClearOS



### Koozali SME Server



### NethServer



### Univention Corporate Server



### Zentyal



Подыскиваем серверные дистрибутивы для некритичных настроек, которыми сможет управлять любой обладающий приемлемыми навыками сетевое управление. Включая домашних пользователей.

# Rosa R9 LXQt

Шашанка Шарму ублажить легко. Нужен лишь настраиваемый дистрибутив с продуманными графическими приложениями для управления системой.

## Вкратце

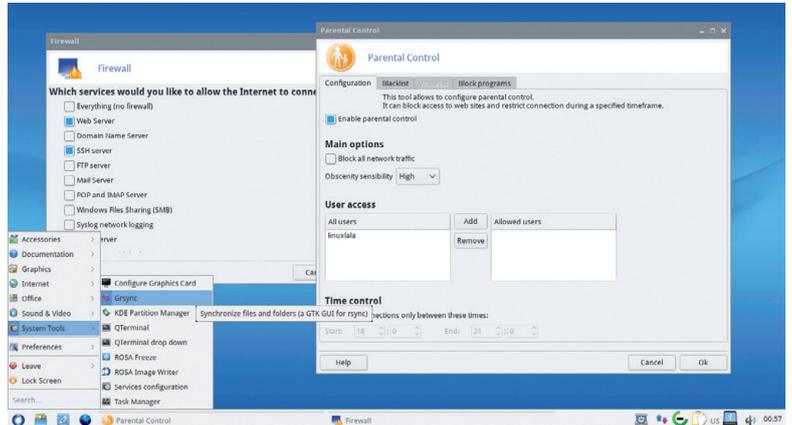
» Для новичков Rosa Fresh LXQt представляет собой простой в использовании настольный компьютер. Стабильность и богатая гибкость его рабочего стола будут привлекать энтузиастов и опытных пользователей, желающих выйти за рамки знакомства с Gnome, Mate и Cinnamon.

Из-за его скромных начинаний в качестве ветки Mandriva, Rosa больше всего известна как дистрибутив KDE, несмотря на то, что дистрибутив теперь разрабатывается в Rosa Labs. Помимо официальных выпусков KDE и Plasma, в проекте также создаются ветки от сообщества, с поддержкой Gnome и LXQt, облегченной среды рабочего стола.

Недавние выпуски других популярных сред рабочего стола вроде Gnome и Plasma почти полностью исключаются на слабых машинах; а преимущество Rosa LXQt заключается в том, что для обеспечения серьезной мощности требуется всего 512 МБ ОЗУ. Rosa R9, поддерживаемая до 2020 г., может стать жизнеспособной альтернативой, если вы устали от Ubuntu и Fedora и похожих на них производных.

## Модная быстрота

Если у вас еще нет доступного раздела на вашем жестком диске, установщик поможет выкроить пространство для Rosa. Помимо обычных вопросов о раскладке клавиатуры, часовом поясе, пароле root и информации для входа пользователя, дистрибутив во время установки не требует никакого другого ввода. По сравнению с другими современными настольными дистрибутивами, весь процесс занимает считанные минуты, даже на машине с одноядерным 2-ГБ ОЗУ. Rosa также не обременяет пользователей сервисами, которые им не нужны, и после установки позволяет вам решить, хотите ли вы, чтобы службы типа CUPS, Samba и OpenSSH запускались во время загрузки.



» Мы обнаружили, что рабочий стол LXQt, рожденный от вдохновенного замеса LXDE и Razor-Qt, не захлаплен и прост в использовании.

Дистрибутив старается быть легковесным, не теряя функциональности. Из всего набора LO Rosa по-прежнему оставляет только LibreOffice Writer и Calc, заменив другие популярные приложения на более разумные альтернативы. Так, вместо Firefox вы найдете браузер Newmoon, потребляющий меньше ресурсов. Rosa также имеет свой домашний Media Player на базе SMPlayer, с поддержкой большинства стандартных медиа-форматов. К сожалению, при обещанной поддержке воспроизведения видео прямо из URL-адресов мы не смогли заставить эту функцию работать.

Загрузчик LXQt позволяет искать приложения и сужает список искомого по мере ввода имени приложения. После запуска приложения, при следующем нажатии на меню, он снова отображает суженный список вместо меню по умолчанию. Раздражает, что для возврата в меню по умолчанию надо нажать Esc: в идеале, строка поиска должна быть сброшена после запуска приложения. Кроме проблемы Rosa Media Player с воспроизведением URL-адресов, это единственная ошибка, которую мы смогли найти в дистрибутиве.

Помимо стандартной поставки интернет- и мультимедиа-приложений, в дистрибутиве также есть несколько полезных приложений, которые помогут вам управлять системой. С удачно названными графическими приложениями, доступными в меню Настройки > Настройки LXQt, можно настроить родительский контроль, брендмауэр и т.д. несколькими щелчками мыши. Пользователям, знакомым с командной строкой, удастся установить тысячи

дополнительных пакетов из своих репозитивов ПО с помощью утилиты urpmi. Но вы можете использовать RPM Drake, графический интерфейс для urpmi, если предпочитаете приложения типа Synaptic.

Wiki-страница на английском языке освещает различные темы для новичков и опытных пользователей. Если вы не нашли руководство по своей проблеме, форумы на английском — отличный ресурс, чтобы обратиться за советом к сообществу.

Этот надежный дистрибутив предназначен для и для опытных, и для начинающих пользователей. Его простой дизайн делает его идеальным для новичков, которые не хотят быть ошарашены незнакомым рабочим столом. И для тех, кто любит возиться со своими дистрибутивами, глубоко настраиваемый LXQt будет отличным выбором. LXF

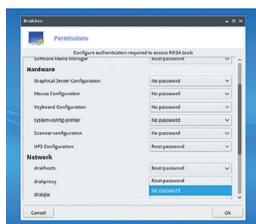


## Свойства навскидку



### Rosa Freeze

Идеален для создания снимка вашей системы, если вы хотите поколдовать с передовым ПО.



### DrakSec

Пригоден для того, чтобы обычные пользователи могли выполнять задачи для администраторов.

## LINUX FORMAT Вердикт

### Rosa Fresh R9 LXQt

Разработчик: Rosa Labs  
Сайт: www.rosalab.com  
Лицензия: GPL и др.

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

» Отлично настраиваемый и потрясающий релиз. Если ищете надежность или производительность, взгляните на него.

Рейтинг **9/10**

# Tanglu 4.0 Gnome

Стараясь не смеяться над трендом давать дистрибутивам странные названия, **Шашанк Шарма** проводит тест-драйв дистрибутива на базе Debian...

## Вкратце

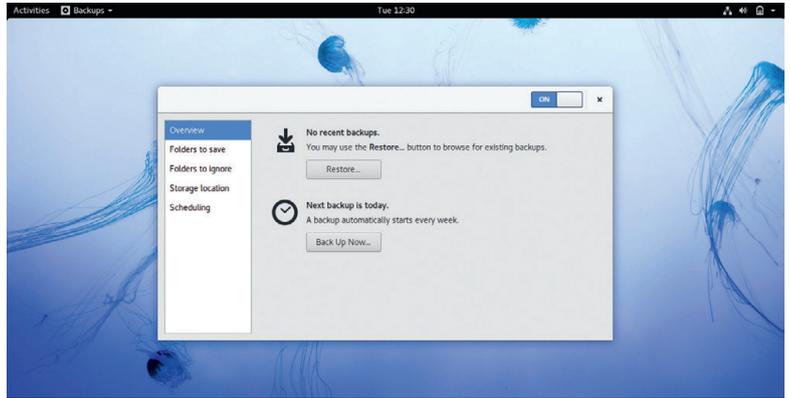
» Дистрибутив на базе Debian порожден разочарованием из-за нечастых выпусков версий Debian. Содержит пакеты из стабильных и тестовых репозиториях Debian. В выпуске с сильной задержкой отображаются Gnome 3.20 и более старые версии большинства ПО. Если вы не возражаете против цикла выпуска, то Debian сам по себе является надежной альтернативой; или попробуйте Linux Mint Debian Edition, если вы предпочитаете возобновляемый дистрибутив.

Увеличение числа дистрибутивов Linux имело негативный побочный эффект, хотя это никогда не обсуждалось и так бы и осталось не высказанным, если бы не Tanglu Linux. При выборе имени проекта разработчики решили, что надо изобрести имя, отвечающее всем критериям: уникальность, узнаваемость, простота запоминания, да чтоб хорошо звучало на английском и немецком языках и бразильском диалекте португальского. В итоге они остановились на Tanglu, производного от Tangerine [англ. мандарин] и Iglou (iglu) [иглу, хижина].

Этот дистрибутив на базе Debian предоставляет 32- и 64-битные ISO-образы с Gnome и KDE Plasma. Для тех, кто презирует графический интерфейс или предпочитает альтернативную среду рабочего стола, есть также редакция Core — минималистский дистрибутив, который пользователи могут формировать по своему вкусу. Также предлагаются контейнер Docker и виртуальная машина для VirtualBox, если вы хотите протестировать дистрибутив, прежде чем прожечь его на жестком диске.

## Кризис самоопределения

Хотя его прародитель Debian пытался выпускать новую версию каждые два года, Tanglu одобряет политику выпуска версии два раза в год. Но это не единственная разница между двумя проектами. Tanglu также поддерживает гораздо больше встроенных программ, чтобы обслуживать большее количество пользователей. По той же причине по умолчанию также включен несвободный репозиторий, который поддерживает драйверы устройств и проприетарные коды.



» Быстрый и проворный дистрибутив лучше всего описать как урезанный Debian, с гораздо меньшим количеством приложений, чем у родителя.

Идея заключается в том, что пользователям не надо прилагать слишком много усилий, чтобы заставить все свои устройства работать с дистрибутивом.

Несмотря на ориентацию на новичков в Linux, примерно 1,3-ГБ дистрибутив предлагает лишь несколько готовых приложений. Ограниченное количество программ переводится как «одно приложение для каждой категории ПО», поэтому в Tanglu вы не найдете несколько браузеров или несколько приложений мультимедиа.

Недостаток Tanglu — это приложение для управления ПО. Установка дополнительного ПО с использованием графического приложения возможна только если вы готовы подбирать его в различных категориях. Дело в том, что внутри приложения клавиатура не функционирует должным образом — некоторые буквы не печатаются, а некоторые повторяются. А значит, вы не можете искать приложения, и навигацию придется выполнять только мышью. Это всего позорнее для выпуска, функции которого просидели в заморозке почти год.

Окончательный релиз вышел почти через год после Бета-версии, и длительная отсрочка была связана с другими обязанностями менеджера проекта. Эта незапланированная задержка означала, что дистрибутив был надолго «заморожен» и, таким образом, имел более старые версии многих приложений. Зная, насколько вредны такие задержки для проекта, разработчики рассматривают решение перейти к плавающему релизу вместо фиксированного цикла выпуска.

Однако формальных решений о будущих выпусках пока не принято. Нынешнее

намерение состоит в том, чтобы иметь расширенный вариант со свежим ПО и параллельный выпуск на базе стабильного Debian.

Проект также будет стремиться к децентрализации власти, чтобы предотвратить повторение узкого места, которое привело к задержке Tanglu 4. Помимо форумов, проект не предлагает какого-либо уникального контента в виде руководств или рекомендаций. Но дистрибутив жаждет предложений по объемам рабочего стола, улучшениям для сайта и т. д.

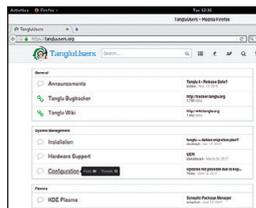
В своей нынешней форме Tanglu не служит какой-либо особой цели, и если проект не наведет порядок в доме, вам будет лучше с любым другим дистрибутивом. Переход на модель с откатным движением приблизит ее к Linux Mint Debian Edition, но отсутствие изюминки будет продолжать мешать внедрению. **LXF**

## Свойства навскидку



### «Голый скелет»

Голый стек приложений на этом маленьком дистрибутиве позволяет добавлять нужные вам инструменты.



### Наследие Debian

Помимо стабильности, линейка Debian означает, что вы легко найдете помощь, если она вам понадобится.

## LINUX FORMAT Вердикт

### Tanglu 4.0 Gnome

Разработчик: Tanglu Project  
Сайт: [www.tanglu.org](http://www.tanglu.org)  
Лицензия: GPL и др.

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

» Желающим принять Tanglu рекомендуем не спешить. Лучше подождите следующего выпуска.

## Рейтинг 6/10

# Robolinux 8.8.1

Шашанк Шарма отказался от Windows, еще учась в средней школе, но Robolinux Cinnamon — дистрибутив для тех, кто так не захотел или не смог.

## Вкратце

» Дистрибутив на базе Debian 8, предназначенный для пользователей Windows. Его изюминка — возможность запускать существующие установки Windows 7 и 10 в качестве виртуальной машины. Как и поддержка, эта функция предлагается только за плату, а не как часть установки по умолчанию. Если вы домашний пользователь или работаете в SOHO, то Linux Mint или Debian с поддержкой сообщества может быть более безопасной альтернативой.

Н ишевые дистрибутивы, нацеленные на пользователей Windows, чтобы помочь тем акклиматизироваться в среде Linux, не новы. Свобода, безопасность, стабильность и постоянно растущая популярность — вполне серьезная причина для того, чтобы заинтересовать пользователей настольных компьютеров, и даже домашние офисы (SOHO, Small office/home office) и малые бизнесы (SMB, Small to Medium Business), попробовать свои силы в Linux. Robolinux, на базе стабильной ветки Debian 8, предназначен для пользователей Windows и пытается представить им дистрибутив без крутой кривой обучения.

Robolinux предлагает и 32-, и 64-битные ISO-образы для всех своих четырех редакций — с рабочими столами Xfce, LXDE, Cinnamon и Mate. К сожалению, его сайт не предоставляет мгновенной ссылки на скачивание, предпочитая спихивать пользователей на свою страницу Patreon, где можно сделать пожертвования на проект, приобрести коммерческое ПО и оплатить техподдержку. Однако страница SourceForge обеспечивает быстрый доступ к загрузке различных выпусков.

Хотя все ISO — это гибридные устанавливаемые Live-образы, заставить их загрузиться с USB-накопителя трудно. Проект рекомендует Unetbootin для создания загрузочного USB-накопителя, но мы не смогли получить результирующий USB-накопитель для загрузки на любом ПК, нетбуке и ноутбуке, приобретенном между 2006 и 2016 гг. Желая попробовать дистрибутив, прожигайте DVD.



» Robolinux поставляется с несколькими приложениями и поддержкой многих карт Wi-Fi, принтеров и графики; и это всё, что здесь хорошего.

## Пожертвование

Robolinux представляет свои установщики «одним щелчком» как уникальную функцию. В меню установщика — длинный список приложений: это такие инструменты, как KeePass, Privoxy, Clam AV, браузер Jon Do и видеоредактор Kdenlive. Судя по названию, можно подумать, что щелчок установщика запускает начало процесса установки, как установщик Steam, поставляемый со многими дистрибутивами. Но только после щелчка по установщику вы узнаете, что они отнюдь не бесплатны. Фактически, пользователи должны внести пожертвование в размере \$10 (£8), которое дает им пакет обновления установщика и дополнительную техподдержку. Не выложив денег, нельзя установить ни одно из приложений, для которых дистрибутив предоставляет установщики «одним щелчком». Кроме того, дистрибутив предоставляет установщики «одним щелчком» для таких популярных приложений, как Wireshark, Torchat и KeePass 2. Они уже являются частью репозитория Debian, но Robolinux не переносит эти пакеты. А значит, желая установить эти приложения, вы должны приобрести обновление установщика приложений. Это отлучение от репозитория Debian, чтобы выдавать приложения только за получив мзду, не только странно, но и откровенно нечестно. Кроме того, проект известен программным обеспечением Stealth VM, которое призвано помочь клонировать диск C:\ из вашей существующей установки Windows и запускать ее как виртуальную машину. Покупка этого приложения после пожертвования \$20 (£16) также дает вам

дополнительную техподдержку по электронной почте и телефону. Понятно, что проект должен зарабатывать на жизнь, для чего и предлагает коммерческую поддержку и программные решения. Но неутешительно, что кроме минималистского FAQ по нескольким вопросам, связанным с установкой, не предоставляется никакой документации или руководств, чтобы помочь пользователям освоить дистрибутив. Нет wiki, списков рассылки или форумов; однако на сайте упоминается, что пользователи не должны платить за техподдержку по запросу, связанным с установкой.

Robolinux — не для домашних пользователей, свидетельством чему — его коммерческие предложения. Мы бы советовали вам воздержаться; ну разве что вы исчерпали все другие дистрибутивы Linux, да так и не нашли себе ничего подходящего. LXF

## Свойства навскидку



### Установщики «одним щелчком»

Это упрощает установку полезных приложений, но только если вы приобрели ПО Installer Upgrade.



### Поддержка Windows

Первоначально доступное только в Robolinux, приложение StealthVM теперь поддерживает большинство дистрибутивов.

## LINUX FORMAT Вердикт

### Robolinux 8.8.1 Cinnamon

Разработчик: Robolinux  
Сайт: www.robolinux.org  
Лицензия: GPL и др.

Функциональность	6/10
Производительность	7/10
Удобство в работе	8/10
Документация	2/10

» Просить покупать коммерческий пакет ради приложений из Debian — это сигнал тревоги. Поищите в других местах.

## Рейтинг 5/10

# AMD Ryzen 5 1400

Зажав свой снобистский процессорный нос, **Джереми Лэйрд** снисходит до нас, простых смертных, по поводу жалких четырех аппаратных ядер.

## Спецификация

- » Разъем AM4
- » Разрядность 64 бит
- » Тактовая частота 3,2 ГГц
- » Турбо 3,4 ГГц
- » Ядра Четыре
- » Потоки Восемь
- » Технологический процесс 14 нм FinFET
- » Кэш L1 384 кБ, L2 2 МБ, L3 8 МБ
- » Память DDR4-2666 (64 МБ макс.)
- » Каналы Два
- » Тепло-выделение 65 Вт
- » Виртуализация AMD-V, AMD-Vi

**Б**удьте осторожны с желаниями. Сегодня мы умоляем Intel не прибедняться. А завтра Intel швыряет на рынок ошеломляющий набор новых процессоров, где наличествует всё — от четырех до 18 ядер. Действительно странные и захватывающие времена.

Пока в воздухе еще витают отголоски Computex 2017, легко упустить из виду менее впечатляющие версии чипа, с которого всё началось. Мы говорим об AMD Ryzen — а конкретно, о Ryzen 5 1400. Сейчас это самый непритязательный Ryzen из всех. Но, рассматривая в контексте, он также и наиболее актуален.

В этом контексте предоставляются четыре надлежащих ядра процессора и поддерживаются восемь программных потоков по цене значительно ниже, чем у Intel. Ryzen 5 1400 укладывается в £160 (\$170). Самый дешевый четырехъядерный чип Intel Core i5-7400 за £165 не кажется отстоящим далеко, но у него всего только четыре потока. Если вам нужно от Intel восемь разблокированных потоков, то это огромный скачок до Core i7-7700K за £325.

Конечно, можно оспаривать значимость технологии одновременной многопоточности, которая позволяет ядру процессора обрабатывать несколько потоков параллельно, но давайте не будем забегать вперед. Сначала рассмотрим некоторые частоты и кэш.

Кроме четырех ядер и восьми потоков, Ryzen 5 1400 отличается тактовой частотой 3,2 ГГц и максимальной частотой Turbo 3,4 ГГц. Как и его четырехъядерный собрат,

он рассчитан на 65 Вт и вставляется в новый основной сокет процессора AMD AM4.

Он также содержит 384 КБ кэша L1 и 2 МБ кэша L2.

Опять же, одинаково с 1500X. Расхождение, где сравнение становится достаточно интригующим, связано с кэшем L3. У 1400 всего 8 МБ по сравнению с 16 МБ у 1500X.

Да, интригует; и не только потому, что у 1500X двойной кэш, а больше кэш-памяти обычно означает большую производительность. Это интригует, потому что способно указывать на внутреннюю структуру двух чипов. В конце концов, кристалл процессора Ryzen состоит из двух четырехъядерных комплектов, у каждого из которых 8 МБ кэша L3. Таким образом, можно подумать, что объем кэша L3 означает работу 1500X в конфигурации 2+2 с включенными обоими комплектами; но в каждом из них включены только два ядра, а 1400 работает с одноядерным комплектом, причем все четыре ядра «отжигают».

Но, похоже, это не так. 1400 работает с той же базовой конфигурацией 2+2, просто некоторые кэши L3 в каждом комплексе объединены. Однако AMD не открывает, поэтому не будет удивительно, если окажется, что она использует оба подхода, чтобы задействовать как можно больше кристаллов.

В конечном счете, всё, что имеет значение — это производительность, и здесь новости в основном очень хорошие. 1400 обеспечивает те показатели, которые вы в целом и ожидаете, основываясь на количестве ядер и потоков и тактовой частоте.

Это означает, что в самых масштабируемых нагрузках, таких как рендеринг, у него есть преимущество над гораздо более дорогими четырехъядерными четырехпоточными чипами, такими как Core i5-7600K, но небольшое отставание, когда дело доходит до программ, чуть больше опирающихся на производительность отдельного ядра. Как и все имеющиеся процессоры Ryzen, 1400 также полностью разблокирован и легко достигает 4 ГГц с самыми элементарными настройками, превосходя частоту модели 1500X, работающую на заводских



» Производительности недорогого чипа AMD хватает, чтобы обогнать свой аналог из Intel.

установках, наверстывая отставание в одном потоке от Core i5, и уж точно уделывает последний в обычной многопоточной пропускной способности.

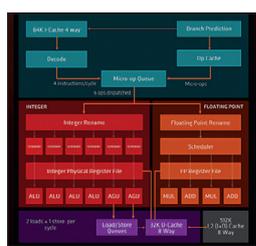
Всё это делает 1400 довольно привлекательной покупкой за его цену. Если и есть проблема, то в том, что чем ниже модели в линейке Ryzen, тем меньше они отличаются. Более того, учитывая, что вам понадобится новая материнская плата и, вероятно, какая-то память, дополнительные £40 за самый дешевый и наш любимый шестиядерный Ryzen 5 1600 выглядят очень даже неплохо. Вы делаете свой выбор, вы тратите свои деньги, и вы не можете сильно ошибиться на этом этапе! **LXF**



## Свойства навскидку



**Охладитель**  
В коробке имеется собственный низкопрофильный кулер AMD Wraith Stealth. Бойтесь и трепещите, хоббиты...



**Архитектура Ryzen**  
Посмотрите на все эти коммуникации! Одна из причин, почему программные потоки Ryzen настолько сильны.

## LINUX FORMAT Вердикт

**AMD Ryzen 5 1400**  
 Разработчик: AMD  
 Сайт: www.amd.com  
 Цена: £160

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

» Лучшее соотношение производительности и цены процессора на рынке. Но бюджетники могут решить подождать Ryzen 3.

**Рейтинг 9/10**

# Dawn of War III

Один-единешенек на сервере мультиплея, **Лейф Джонсон** услышал клич: «Молот, за дело!». По счастью, молот у него был с собой...

## Спецификации

### Минимальная:

- » **ОС** Ubuntu 16.04 64-бит, SteamOS 2.0
- » **Процессор** 3,4 ГГц Intel Core i3-4130
- » **ОЗУ** 8 ГБ
- » **GPU** 1 ГБ Nvidia 650Ti, 1 ГБ VRAM (драйвер 375.66)
- » **На диске** 30 ГБ

### Рекомендуемая:

- » **ОС** Ubuntu 16.10 64-бит, SteamOS 2.0
- » **Процессор** 3,4 ГГц Intel Core i7-4770
- » **ОЗУ** 8 ГБ
- » **GPU** Nvidia 980Ti (Драйвер 375.66), 4 ГБ VRAM (AMD и Intel на момент релиза не поддерживаются)
- » **На диске** 30 ГБ
- » **Vulkan** Nvidia драйвер 381.22, Mesa 17.1 для AMD, Mesa 17.2-devel для Intel



» В битве элитные единицы как боги, и не отделаться от мысли, что они слишком сильны.

**И**нопланетный пафос в *Warhammer 40 000: Dawn of War III* прет из каждого пикселя, так что нам было просто не устоять. Перед нами открывается будущее настолько продвинутое, что полет на космическом корабле для людей — всё равно что для нас поездка на автобусе, и всё же одним из его величайших героев здесь является парень, орудующий огромным молотом.

Зато как орудующий! Зовут его Габриэль Анджелос [Gabriel Angelos], он командир Кровавых воронов [Blood Ravens] из отряда Космических десантников, и может раскидать целую ораву орков не хуже, чем Крис Фарли [Ред.: — Кто-кто?], запустив в них кофейным столиком из бальсы.

Еще одно нажатие горячей клавиши, и его молот начнет вращаться на 360 градусов, так что кровь орков разольется повсюду, такое жутковатое подражание Джексону Поллоку. От последнего вы в этой долгой серии стратегий реального времени (RTS) по франшизе Relic Entertainment, возможно, подустали, но всё-таки сделаны они со вкусом, от и до.

*Dawn of War III* всячески стремится приблизиться к классике данного жанра. И по большей части игре это удастся. Не только за счет таких (буквально) колоссальных личностей, Габриэль, но и за счет многотысячных, не уместящихся на экране армий, ограниченным базовым набором зданий и стычками за ресурсы. Кое-что

из старого арсенала элементов в процессе порастерялось — та же система укрытий и охота за награбленным в духе *Diablos*, которые сделали *Dawn of War II* столь увлекательной в 2009 г.; но главное, что вам светит в этом завтрашнем дне, это неустанный и агрессивный экшн. Некоторое однообразие в сценариях здесь компенсируется насыщенностью действия, так что скучать в многопользовательском режиме или при прохождении кампании нам не пришлось.

Но будь сюжет получше, *Dawn of War III*, возможно, даже была бы великолепной. Все компоненты для этого здесь налицо, но они растрачиваются на слабо цепляющую историю про людей, орков и эльдаров, упрямо ведущих непримиримую войну за таинственное копье и ведение дел на планете-беглянке. И война! И предательство! И странные альянсы! Иначе выражась, типичный *WarHammer*.

Другое дело, что *Dawn of War III* позволяет играть во все три фракции в кампании с 17 миссиями, не ограничивая себя Космическим десантом. У каждой группы очень ощутимая своя специфика, и возможностью с ней поэкспериментировать *Dawn of War* увлекла нас больше всего.

Уследить за всеми боевыми подразделениями бывает нелегко, особенно когда



» Давно мы так не отводили душу в реальной боине в стратегии реального времени.

экран оказывается заполнен маленькими зелеными человечками; но интуитивный интерфейс пользователя обычно сглаживает потенциальные проблемы. Для каждого отряда, находящегося или готовящегося выйти на поле боя, отведен маленький квадрат в нижней части экрана; это позволяет легко отслеживать, какие из них были жестоко обстреляны и нуждаются в поправке здоровья.

Однако из всего того креатива, что ушел на придание своеобразия фракциям, мало что досталось собственно картам. Типовая миссия — вы убиваете грохочущие сапоги какого-нибудь элитного подразделения вроде Горгутца, и вашим бойцам придется пробивать себе путь из одной точки в другую и собирать узлы ресурсов по всей карте. Затем мы создали несколько базовых зданий, таких как казармы и магазин новейшей техники, а затем приняли бой с очередным противником. И на этом, к сожалению, и всё.

Но так или иначе, в RTS скучная кампания — это ожидаемо. В конце концов, есть же такая штука-с-долей-шутки, что это просто длинные учебники по прохож-

## Иногда матчи сводятся к тому, что две элиты мутузят друг друга.

дению многопользовательского режима. А в чем подвох? *Dawn of War III* тут подешевил. Вместо того, чтобы отправлять вас в несколько миссий подряд с одной и той же фракцией, Relic каждый раз дает вам новую.

Это лучше переваривалось бы, если бы фракции действовали чуть более схожим образом; но всё там настолько по-разному, что трудно разобраться с такими тонкостями, как управление обновлениями для отрядов, специфические умения трех элитных героев и их пехотинцев и порядок построения конкретных зданий. Только освоишься с тактикой одной фракции, как надо переключаться на другую. Всё это идет в ущерб логике повествования, так как



► **Видали, какие пушки?**

бессвязный сюжет не дает сконцентрироваться на одной фракции.

Тень *Warcraft III* нависает над героями так сильно, что сам Габриэль, возвышаясь среди всех своих соплеменников со своим двуручным молотом, легко сошел бы за современную модель для скульптуры Артаса Менетила [Arthas Menethil]. Его манера сражаться с орками скорее напоминает комикс, чем мрачное фэнтези, а актер, который его озвучивает, явно нравится с чувством выкрикивать эти дурацкие пафосные реплики. (Звуковое оформление, в целом, очень сильное).

Обычно больше всего удовольствия от игр RTS мы получаем, грамотно выстроив серию войск Бугавей [Grunts], но здесь все эти премудрости оборачиваются пустой тратой времени, поскольку элиты разбивают ваши армии за пару секунд. Технически таких героев даже нельзя убить. Сначала вы формируете элитные отряды разной силы, накапливая очки в ходе пассивной игры (за одного Да Морканоб [Da Morkanaut] можно получить целую девятку), но когда они погибают, вам надо только немного подождать, и потом они снова смогут крушить всё направо и налево. На самом деле, их роль в стратегии так велика, что в многопользовательском режиме мы частенько наблюдали, как игроки просто ползают вокруг своей базы, накапливая очки, чтобы бросить на поле боя какой-нибудь элитный отряд — а что тут такого?

► **Имперский рыцарь готовится выйти на поле боя.**

Как насчет самого режима? Он тоже возвращает нас к классическим стратегиям в реальном времени, где всё строится на создании баз, выкачивании ресурсов и объединении армий людей, орков и эльфов вместе в одну гигантскую разношерстную орду в режиме 3-на-3, 2-на-2 или 1-на-1.

Наблюдаются и довольно ощутимые новшества. В битвах главное — не разрушение баз, а скорее захват «силовых точек»... но для начала нужно будет справиться

с генераторами защит и пушками. Короче, на деле многопользовательский режим в *Dawn of War III* очень похож на боевую арену в духе *League of Legends* или *Dota 2*.

Наличие элит и таймеров эскалации означает, что умные игроки всегда находятся в наступлении, держась подальше от элит и постоянно пытаюсь нанести урон силам противника. Больше всего действия сосредоточено вокруг узлов ресурсов, поскольку их потеря равна удару, от которого фракция уже не оправится. Играете на оборону? Вероятно, вы получите несколько неприятных комментариев в чате от своих соратников за то, что вы им не помогли.

Элиты настолько доминируют на поле, что мы наблюдали, как целые матчи превращаются в то, как две элиты просто мутузят друг друга, а вокруг лежат мертвые армии. Именно в такие моменты они больше всего напоминают подразделения героев MOBA, и выглядит это весьма неприглядно.

Есть недостатки и в самом ходе игры, как то: отсутствие автосохранения во время длительных миссий кампании или невозможность изменения ключевых слов. Но, при всем при этом, играли мы всё же с удовольствием. LXF

## LINUX FORMAT Вердикт

### Dawn of War III

Разработчик: Feral Interactive  
Сайт: www.feralinteractive.com  
Цена: £40

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

► Стоит похвалить *Dawn of War III* за попытку обновить формат. И в лучшие моменты игра прямо-таки захватывает.

**Рейтинг 7/10**





# Мобильные НОВОСТИ

## МОБИЛЬНАЯ СВЯЗЬ

# На пути к 5G

Verizon, Ericsson и Qualcomm преодолели барьер 1 Гбит/с.

**L**TE-модем Qualcomm Snapdragon X20 обеспечил скорость передачи данных 1,07 Гбит/с. Передача осуществлялась в принадлежащей Verizon коммерческой сети Gigabit LTE с использованием всех лицензируемых частотных диапазонов, двенадцатью одновременными LTE-потоками, с агрегацией трех несущих при ширине 20 МГц в FDD-диапазоне; для каждой несущей задействовались модуляция 256 QAM, оборудование 4×4 MIMO и множество антенн на вышках сотовой связи и оконечных устройствах.

Новый импульс получила проводимая Qualcomm Technologies стратегия, направленная на продвижение LTE в качестве стандарта беспроводной широкополосной связи ближайшего будущего (согласно опубликованному в июне отчету Ericsson и Nokia, несмотря на нарастаемые в мире

усилия по созданию сетей 5G, к 2022 г. ими будут охвачены не более четверти абонентов мобильной связи в Северной Америке, и еще меньше в Европе и странах АТР).

Qualcomm уже длительное время рекламирует свой Snapdragon X20 в качестве обо-

## Сделан важный шаг на пути к 5G для всех абонентов.

рудования "gigabit LTE", однако в реальных условиях мобильные операторы заявленной скорости достичь не смогли. «В сотрудничестве с Verizon и Ericsson мы первыми преодолели барьер скорости в 1 Гбит/с. Сделан важный шаг на пути к 5G, обеспечивающий более высокие средние скорости



» «Большая тройка» Verizon, Ericsson и Qualcomm доказала живучесть текущей версии широкополосного стандарта LTE.

передачи данных для всех абонентов и выводящий пользовательский опыт на новый уровень», заявил представитель Qualcomm. Впрочем, скептики уже прокомментировали, что имело место «всего лишь одно-разовое достижение, которое невозможно воспроизвести вне лаборатории».

## ВИЗУАЛЬНЫЕ ПРОЦЕССОРЫ

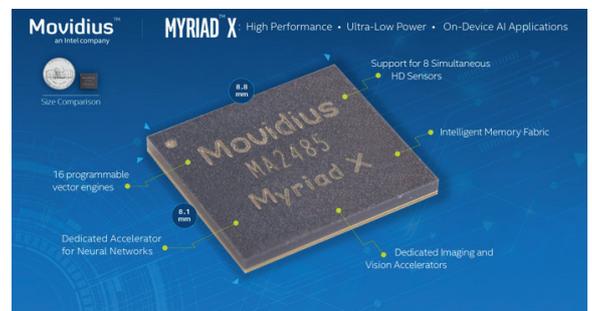
# Первый со своей нейросетью

Intel представила VPU Movidius следующего поколения.

**В**изуальные процессоры (VPU) нового поколения Movidius Myriad X, о выпуске которых Intel объявила в конце августа, предназначены для организации машинного зрения автопилотируемых автомобилей, роботизированных устройств и т.п. систем. Myriad X является первой в мире однокристальной системой (SoC), оснащенной специализированным аппаратным блоком Neural Compute Engine, который разработан для взаимодействия с нейронными сетями глубокого обучения. При крохотных размерах (8,5×8,7 мм), Myriad X показывает 10-кратный рост пиковой производительности в приложениях с одновременным функционированием нескольких нейросетей, при этом его энергопотребление (0,5 Вт) осталось на уровне предыдущей модели. Производительность

блока Neural Compute Engine при взаимодействии с DNN (deep neural network) достигает 1 TOPS (триллион операций в секунду), а суммарная производительность Myriad X составляет 4 TOPS.

Myriad X оснащается 16-ю 128-разрядными векторными процессорами VLIW, и допускает подключение по 16 линиям MIPI до восьми HD-камер с разрешением 720p (RGB) и частотой до 180 Гц, с параллельной обработкой всех потоков; поддерживаются стандарты LPDDR4, USB 3.1, PCIe 3.0, кодирование 4K-видео с частотой 30 Гц (H.264/H.265); суммарная пропускная способность VPU Myriad X составляет 700 МПикс/с. Объем встроенной памяти — 2,5 МБ, архитектура Homogenous On-Chip Memory обеспечивает пропускную способность 450 ГБ/с.



» Высокая производительность при миниатюрном форм-факторе позволяет Intel рекламировать Myriad X как идеальный визуальный процессор для автономных платформ.

Myriad X поставляется вместе с арсеналом разработчика. SDK включает нейронный сетевой компилятор и специализированный фреймворк FLIC.

## БЕСПИЛОТНЫЕ АВТО

## Я буду долго гнать... автомобиль

Intel и Mobileye создают тестовый автопарк из автомобилей уровня L4.

В августе израильская компания компьютерного зрения Mobileye стала частью корпорации Intel (о сделке на сумму \$15,3 млрд было объявлено в марте), и приступила к созданию парка автомобилей четвертого уровня автономности (L4 по классификации SAE); автомобиль с таким уровнем автоматизации способен самостоятельно справляться с большинством ситуаций на дороге. Первые машины будут готовы до конца текущего года, после чего начнется их тестирование на автострадах США, Израиля и стран Европы; всего же парк будет состоять из более чем сотни полностью автономных автомобилей. Технологически Intel создает целостную систему “car-to-cloud”, объединяя свои высокопроизводительные компьютерные ноу-хау и технологии Mobileye: компьютерное зрение, сбор и обобщение данных с различных датчиков, картографирование, машинное обучение, искусственный интеллект.

«Создание автомобилей и их испытание в реальных условиях обеспечивает оперативную обратную связь и ускоряет внедрение технологий и решений для автомобилей высокой степени автономности и для полностью автономного транспорта, — заявил будущий старший вице-президент Intel и будущий CEO/CTO Mobileye Амнон Шашуа [Amnon Shashua]. — Географическое разнообразие при проведении испытаний необходимо, поскольку разные регионы имеют сильно различающиеся стили вождения, различные дорожные условия, знаки и дорожную разметку. Наша цель заключается в разработке технологий беспилотных автомобилей, которые можно применять повсеместно; следовательно, нам необходимо тестировать и „обучать“ наши автомобили в самых разных местностях».

Испытательный парк будет состоять из нескольких марок и типов автомобилей, чтобы продемонстрировать универсальность технологии. Intel пытается создать



► CEO Intel Брайан Кржанич [Brian Krzanich] и соучредитель Mobileye Амнон Шашуа (справа) выступают на пресс-конференции BMW Group.

платформу, разворачиваемую и масштабируемую в условиях реального мира, которую автопроизводители смогут приобрести и настроить под конкретные условия с помощью собственного ПО для автоматического вождения.

## CPU

## «Кофе» подадут потом

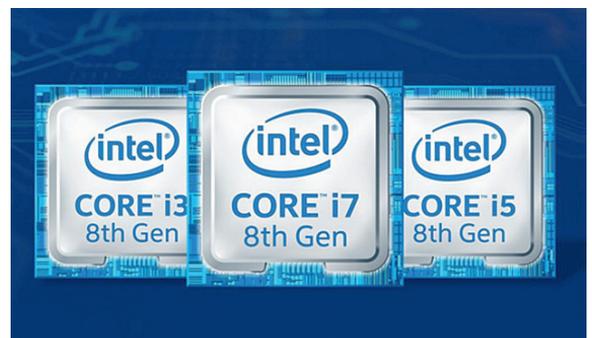
VIII поколение CPU от Intel для ноутбуков повысит производительность на 40%.

Intel представила 8-е поколение линейки процессоров Core. По непонятной пока причине, новые 15-ваттные CPU Core i3, i5 и i7 выполнены на обновленной архитектуре Kaby Lake Refresh; выпуск процессоров широко разрекламированной архитектуры “Coffee Lake” Intel обещает начать позже в этом году. Первые 4-ядерные CPU «серии U» предназначены для ноутбуков и ультрабуков; в состав линейки входят модели i7-8650U с базовой частотой 1,9 ГГц и предельной частотой разгона (boost clock) 4,2 ГГц, i7-8550U (1,8 ГГц и 4 ГГц), i5 8450U (1,7 ГГц и 3,6 ГГц) плюс i5-8250U (1,6 ГГц и 3,4 ГГц).

Intel утверждает, что ее обновленная архитектура 14+ нм обеспечивает 40-процентное повышение производительности по сравнению с предыдущим поколением процессоров Kaby Lake, и обещает, что обновляющиеся с компьютеров пятилетней давности увидят «двойное повышение

производительности». Директор по маркетингу мобильных платформ Intel Карен Регис [Karen Regis] назвала новую линейку «огромным скачком, происходящим, возможно, один раз в десять лет. Новые чипы затмевают в отрасли всё остальное» (более чем прозрачный намек на линейку AMD Ryzen). Как заявила г-жа Регис, пользователи новых процессоров увидят 30-процентное улучшение скорости при редактировании изображений, 40-процентное — в многозадачности, и 48-процентное улучшение скорости при создании слайд-шоу в PowerPoint.

Несмотря на значительное увеличение производительности, время автономной работы, по словам представителя Intel, остается «бескомпромиссным»: устройства с CPU VIII поколения обеспечат до 10 часов автономной работы при просмотре локального видео 4К, и до 11 часов для потокового контента 4К с YouTube.



► Intel обещает, что уже в сентябре можно будет приобрести более 145 OEM-устройств с CPU нового поколения — но пока еще не «Coffee Lake»...

Кроме того, обновленная архитектура Kaby Lake поддерживает смешанную реальность Windows Mixed Reality, аутентификацию через Windows Hello и fingerprint touch-to-pay, а также протокол Thunderbolt 3.

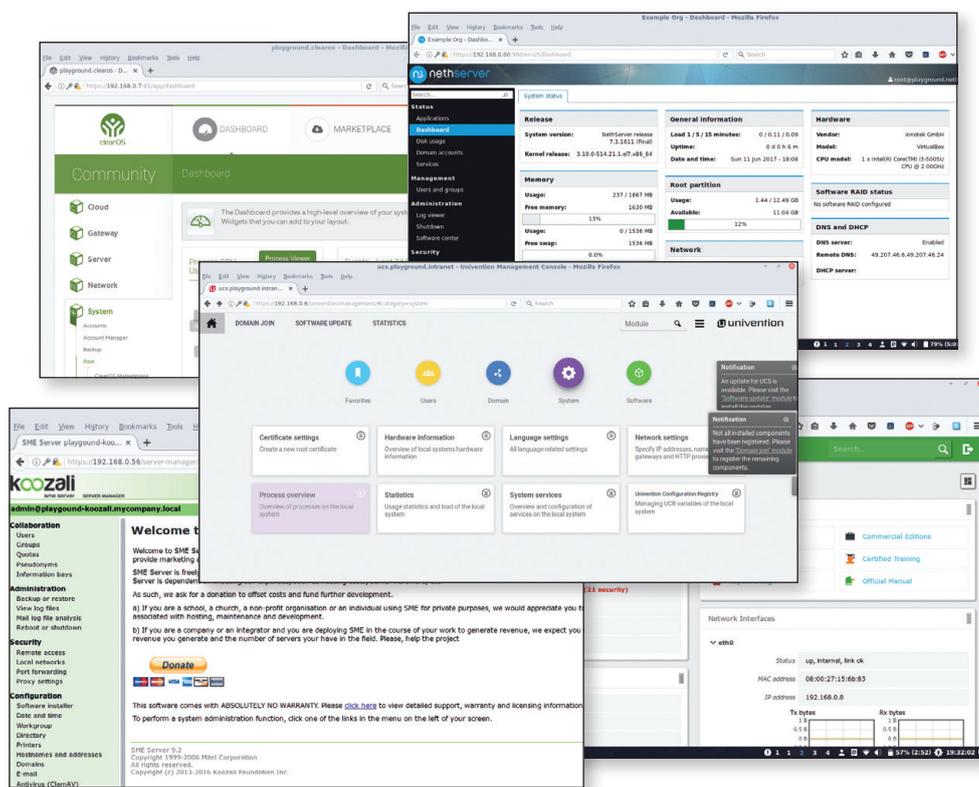
Интегрированная в процессоры U-серии графика HD 620 также обновлена до UHD 620; GPU в моделях Core i5 работают на частоте 1,1 ГГц, в Core i7 — 1,15 ГГц. LXF

# Сравнение

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

## Бизнес-серверы

Идеал дистрибутива для сервера — баланс стабильности, функциональности и адаптируемости. Маяк Шарма выбирает наилучший именно для вас...



### Про наш тест...

Бизнес-дистрибутив для сервера должен предлагать мониторинг сетевой инфраструктуры, управление доступом к Интернету, разделение ресурсов между пользователями и т.д. И это помимо его более общей функции управления web, почтовым или файловым сервером. В идеале дистрибутив не должен заставлять пользователя совершать слишком много шагов для завершения установки, и не требовать слишком много времени или усилий для настройки его разнообразных сервисов.

Помимо акцента на настройке и управлении, дистрибутив должен вовремя обновляться и исправлять ошибки. Конечные потребители здесь малые/домашние офисы, поэтому доступность документации и поддержки — тоже важный фактор. Поскольку больше дистрибутивов предлагают 32-битные релизы, мы сравнили эти дистрибутивы на 64-битном компьютере, запуская слегка устаревшее ПО во избежание проблем с надежностью.

**С**табильность и надежность дистрибутивов Linux может привести некоторых к мысли, что любой старый добрый дистрибутив Linux годится для работы со сложными серверами; однако это не так. Для начала, у настольного дистрибутива есть масса компонентов, просто не относящихся к серверу. Их замена на важные серверные приложения — задача не из простых, и лучше всего будет использовать дистрибутивы, для серверов и разработанные.

Но развертывание и настройка сервера — трудоемкий процесс. Если вы готовы частично поступить контролем над выбором программ и настроек, есть специально

настроенные дистрибутивы, собранные профессионалами, которые выполняют ряд серверных задач. Для нашего Сравнения мы отобрали дистрибутивы, развертывающие сложный сервер через интерфейс наведи-и-щелкни.

Простота администрирования, скорая готовность к работе и обширный список функций представляют собой только верхушку айсберга, которым является бизнес-

дистрибутив, призванный к обслуживанию разнообразных пользователей в среде малого/домашнего офиса и не подпадающий под стереотипное восприятие ограниченности. Мы ищем серверные дистрибутивы для некритичных настроек, подвластных любому освоившему сносные навыки сетевого управления. Вот почему Debian и CentOS не вошли в нашу подборку, хотя это превосходные дистрибутивы для сервера.

### Наша подборка

- » ClearOS
- » Koozali SME Server
- » NethServer
- » Univention Corporate Server
- » Zentyal

**Мы отобрали дистрибутивы, развертывающие сервер через интерфейс наведи-и-щелкни.**

# Орудия труда

Какие предлагаются функции?

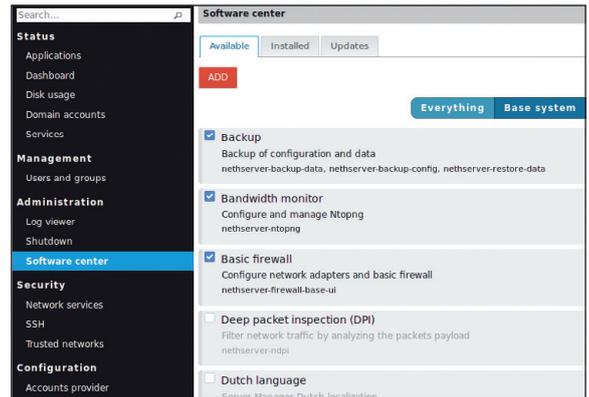
**С**ервер Linux может означать компьютер, обслуживающий веб-страницы, файлы и электронную почту, или брандмауэр или шлюз, защищающий ваши сетевые компьютеры от угроз из Интернета. Koозali можно настроить как сервер шлюза, если вы хотите, чтобы он работал как брандмауэр и обеспечивал внешние почтовые и веб-серверы. Его также можно настроить в качестве частного сервера шлюза, в этом случае он работает как брандмауэр, но отключает весь входящий трафик.

Третий вариант — настроить его в режиме только сервера. Если вы выберете последнюю опцию, вас попросят настроить также отдельный сервер шлюза, чтобы он работал в качестве брандмауэра, если вашему серверу нужно подключение к Интернету. Этот дистрибутив к тому же позволяет создавать резервные копии и способен восстановить вашу предыдущую установку Koозali.

Zentyal пригоден для настройки домена и сервера директории. В качестве почтового сервера он предоставит фильтр почты и сервис сканирования на вирусы. Вы также

можете использовать VPN и Webmail независимо от сервисов инфраструктуры, таких как работающий IM-сервис, DNS- и DHCP-серверы, в дополнение к выпуску и управлению сертификатами безопасности. В отличие от Zentyal, NethServer можно также применять для всевозможных задач сервера. Вы можете использовать его веб-интерфейс для формирования базовой установки. С помощью Software Centre легко конвертировать базовую установку NethServer в веб-сервер Apache, файловый сервер, основанный на XMPP сервер обмена мгновенными сообщениями и/или почтовый сервер, настроить подключение OpenVPN и даже сервер ownCloud всего за несколько щелчков мышью.

Univention, как и Zentyal, разработан строго для среднего и малого бизнеса и/или предприятий. Он умеет запускать принтер и файловый сервер, а также веб-прокси с возможностью сканирования на вирусы, и пригоден для настройки почтового сервера и даже включения сервисов совместной работы. Можно использовать App Center для добавления большего количества пакетов и сервисов.



Интерфейс NethServer дополняется кучей сервисов, благодаря чему подходит для множества задач.

ClearOS отличается от остальных тестируемых в нашем Сравнении дистрибутивов подборкой поддерживаемых им серверных программ. ClearOS несложно настроить для работы в качестве сетевого сервера, сервера шлюза или облачного сервера в дополнение к обычным серверам, таким как сервер директории, сервер базы данных, почтовый сервер, веб-сервер, FTP сервер, фильтр контента, сервер печати через Samba или CUPS и т. д. Подборка инструментов управления системой и сетью в ClearOS может использоваться для создания резервных копий и списков контроля доступа, управления пропускной способностью и других подобных задач.

**Вердикт**

ClearOS ★★★★★  
 NethServer ★★★★★  
 Koозali ★★★★★  
 Univention ★★★★★  
 Zentyal ★★★★★

» Все серверы выиграют от добавления функций, если не полениться это сделать.

# Стабильность и безопасность

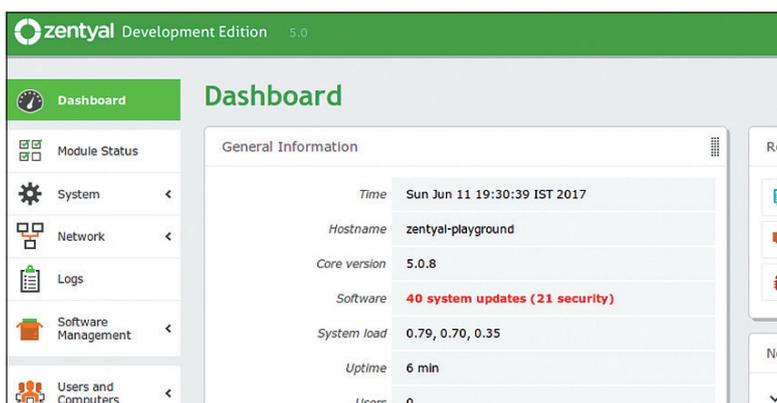
Получают ли программы регулярные релизы и частные обновления?

**П**ри выборе серверного дистрибутива необходимые требования — стабильная основа и тщательная система сообщений об ошибках и их отладках. Но даже это — всего лишь часть уравнения. В идеале дистрибутив должен постоянно выпускать отладки и обновления

безопасности, обеспечивая вашему серверу максимальный срок службы.

Произошедшие от CentOS, NethServer, ClearOS и Koозali зависят от вышестоящих релизов и не имеют фиксированной схемы выхода релизов. Основа Koозali — CentOS 6.7, а остальных — CentOS 7. Релизы Zentyal

основаны на самом свежем релизе Ubuntu LTS. Community Edition [Редакция Сообщества] служит тестовым стендом для технологий, которые войдут в коммерческое предложение. Свободный вариант предлагает разработанные сообществом, нетестированные обновления, а официальные и тщательно протестированные обновления резервируются для коммерческих редакций. ClearOS точно так же предлагает программы проверенного качества в своей Бизнес-редакции. Однако обновления от сообщества должны удовлетворить потребности большинства настроек для малого и домашнего офиса. Благодаря своей основе — Debian, Univention стабилен и надежен. Обновления безопасности для каждого релиза выходят в течение не более полугодия после осередного релиза. Для сравнения, коммерческий вариант даст вам 5 или 7 лет поддержки в зависимости от модели оплаты. Zentyal будет поддерживаться до 2021 г., ClearOS Community — до марта 2019 г., а Koозali — до ноября 2020 г.



Как и лежащий в его основе дистрибутив, Zentyal быстро публикует обновления и отладки.

**Вердикт**

ClearOS ★★★★★  
 NethServer ★★★★★  
 Univention ★★★★★  
 Koозali ★★★★★  
 Zentyal ★★★★★

» Срок поддержки у Zentyal больше, но ему не хватает протестированных обновлений.

# Интерфейс управления

Легко ли будет разобраться с инструментами и функциями?

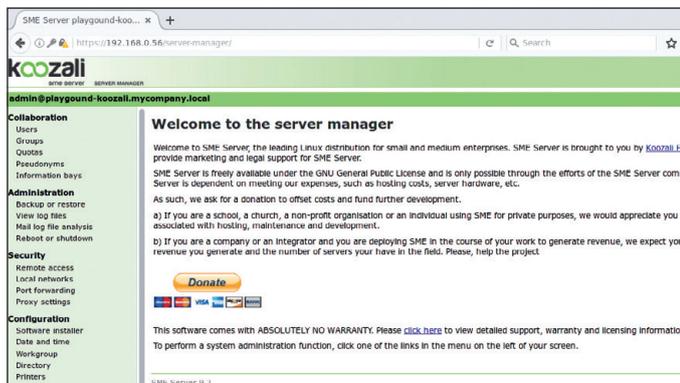
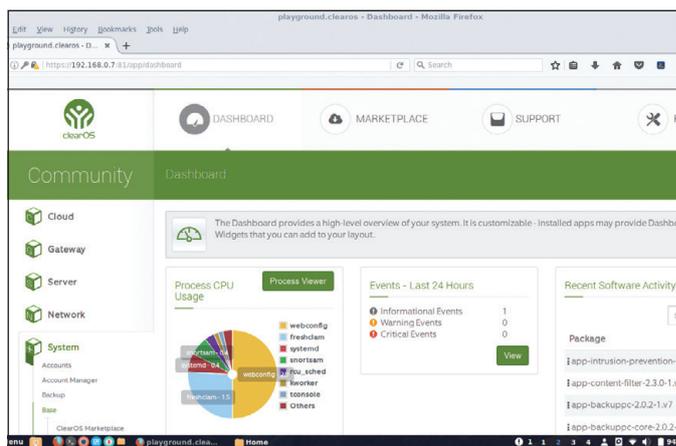
**У** всех дистрибутивов простая установка и почти одинаковый набор функций. Но после установки вам всё равно надо настроить дистрибутив, чтобы он соответствовал вашим потребностям. Еще важнее то, что управление и мониторинг бизнес-дистрибутива — это процесс

постоянный. Если только вам не доставляет удовольствие возня с файлами настройки в терминале с использованием *emacs* или *vim*, вам захочется использовать дистрибутив, который предлагает хороший интерфейс администратора. Это особенно существенно на серверах без клавиатуры, мыши

и монитора, которые управляются через другие компьютеры в сети. Обычный сервер использует множество четко определенных сервисов и программ. Вот почему важно, чтобы интерфейс был организованным и интуитивным, чтобы вы могли разобраться в нем быстро и без проблем.

## ClearOS ★★★★★

Основанный на web Dashboard по умолчанию кажется пустым, но туда можно добавить элементы, например, Недавно использованные программы [Recent Software Activity] и Потребление CPU [CPU Usage]. Многие дополнительные приложения также обеспечивают виджеты, которые можно добавить в dashboard. Нажмите на вкладку Support [Поддержка], если вам нужна помощь в понимании концепции или вы столкнулись с ошибкой и хотите о ней сообщить. Marketplace дает доступ ко всем имеющимся программам, которые вы можете установить, включая такие категории, как Cloud [Облако], Server [Сервер] и Network [Сеть]. Вы можете выбрать просмотр только бесплатных или только платных программ и даже ограничить выбор определенным промежутком времени. Из каждого экрана web-интерфейса вы можете получить доступ к документации, нажав на кнопку Documentation. Например, при нажатии на Gateway > Proxy > Web Proxy Server [Шлюз > Прокси > Прокси-Сервер Web] на боковой панели и затем на Documentation [Документация] откроется руководство по настройке.



## Koozali ★★★★★

Когда экран входа вместо «здрасьте» просит денег, проект это не очень-то красив. Некоммерческий фонд Koozali Foundation зависит от пожертвований, чтобы продолжать предлагать дистрибутив бесплатно. Менеджер программ — самое плохое, что здесь есть. Обновления и добавочные программы для установки предлагаются в списке прокрутки — искать пакеты нельзя. Обновления можно установить выборочно, а не всё разом. Все настраиваемые элементы перечислены под своими заголовками: например, Administration [Администрирование], Security [Безопасность] и Configuration [Настройка]. Можно задать квоту на диске, через Collaboration > Quotas [Сотрудничество > Квоты], или сделать резервные копии данных сервера (файлы настройки и данных пользователя) на накопитель или локальный ПК. Нажмите на Backup [Резервное копирование] или выберите нужную опцию в Administration [Администрирование].

# Документация и поддержка

Для тех случаев, когда нужна рука помощи...

**З**накомство с устройством Linux и навыки работы в сети — вот, может быть, и всё, что вам понадобится для установки сервера; но установленный дистрибутив — это еще не сервер. Даже опытному оператору может понадобиться помощь, чтобы заставить все сервисы ладить друг с другом.

Вот почему документация является важным фактором в выборе серверного дистрибутива. Все наши конкурсанты предлагают похожий набор помощи для пользователей. Каждый из них имеет форумы —

это всесторонние репозитории подсказок и советов по настройке для разных сервисов и наилучшей работы безопасности. Даже краткое время, проведенное на форуме, выявит, как избежать многих подводных камней. Есть руководства в виде wiki и How-To, а также руководства для пользователей и администраторов.

Для своей редакции Сообщества Univentio предлагает форумы и подробные руководства по разным вопросам, снабженные скриншотами. Koozali позиционирует себя почти как сервер для новичков.

Помимо руководств для администратора, пользователя и разработчика, с инструкциями по разным аспектам сервера, wiki заодно размещает большое количество справочников практически по всем вопросам. Если вам нужна более конкретная помощь, отличным местом будут форумы для общения с сообществом.

NethServer, ClearOS и Zentyal предлагают практически одно и то же, но Zentyal их опережает со своими Zentyal Server Book и онлайн-курсом для сетевых администраторов, стоимостью около £435.

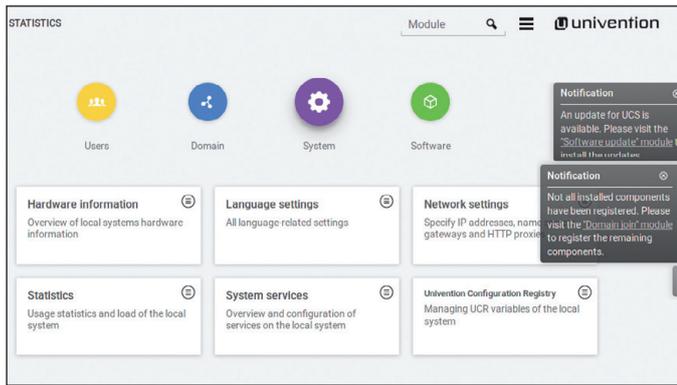
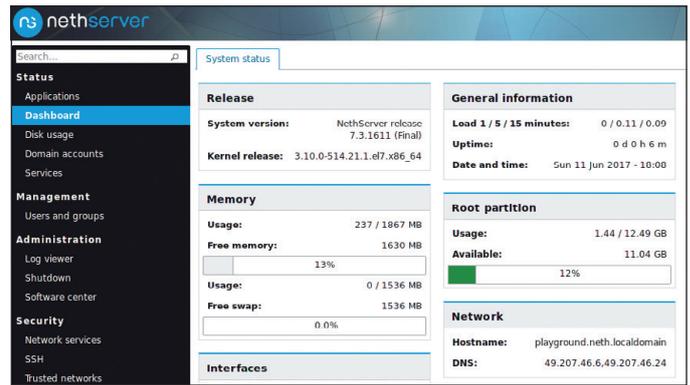
## Вердикт

- ClearOS ★★★★★
- Koozali ★★★★★
- NethServer ★★★★★
- Univentio ★★★★★
- Zentyal ★★★★★

» Справочники являются смесью вводных статей и пошаговых руководств.

## NethServer ★★★★★

Вы оцените рвение NethServer по части безопасности. Он настаивает на пароле, имеющем в составе заглавные буквы и спецсимволы, а также предлагает отключить DHCP, и позволяет выбрать IP-адрес во время самой установки. В установке по умолчанию программ нет вообще, и вы должны установить их с помощью Software Center. Доступные приложения делятся на категории. Нажмите на Base system [Базовая система], если хотите установить Firewall [Брандмауэр], Email [Почтовый], File server [Файловый сервер]. Программы для совместной работы и менеджер виртуальной машины находятся в категории NethForge, а сетевые утилиты, такие как монитор пропускной способности и система предотвращения вторжений — в Firewall. Как и в других дистрибутивах, систему нельзя использовать, пока скачивается и устанавливается программа. Все установленные приложения сгруппированы во вкладке Applications.

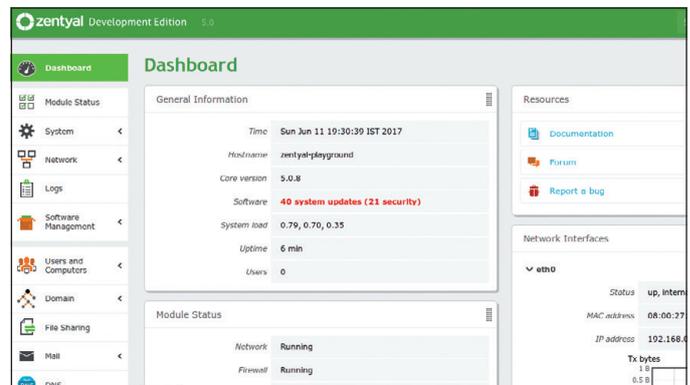


## Univention Corporate Server ★★★★★

Univention не использует боковой панели, чтобы помочь вам в навигации. Его панель управления объединяет в середине экрана ряд ярлыков для разных настраиваемых элементов, таких как Users and Software [Пользователи и Программы]; нажатие на каждый отображает дальнейшие опции. Интерфейс предполагает, что пользователи знакомы с используемой концепцией, и не пытается свести все опции к наведи-и-щелкни. Большая часть опций настройки вынесена в ярлык System на панели управления: Network settings [Настройки Сети], System services [Системные Сервисы] и т.д. Можно запускать и останавливать сервисы вроде Apache и ClamAV, войдя в System > System services [Система > Системные Сервисы]. Появится список всех сервисов. Нажмите на окно рядом с сервисом для доступа к кнопкам Start, Stop и Restart. Также можно щелкнуть правой кнопкой по имени сервиса и выбрать требуемую операцию.

## Zentyal ★★★★★

Zentyal загружает минималистичную графическую среду, где на странице логики на открыт Firefox. Если вы не планируете администрировать установку Zentyal локально, вы, вероятнее всего, будете использовать web-интерфейс, получая доступ к нему с отдельного компьютера в сети. При установке пакетов интерфейс не упоминает общего размера скачанных зависимостей. В результате установка даже трех сервисов, таких как брандмауэр, web-прокси и web-почта, может потребовать более 400 пакетов. После установки вы сможете включать, выключать и настраивать эти сервисы из самого web-интерфейса. Панель управления, как и левая боковая панель, заполняются по мере установки вами большего количества сервисов. При внесении изменений в сервис сначала нажмите на кнопку Change [Изменить], а затем на кнопку Save Changes [Сохранить изменения], чтобы изменения вступили в силу.



# Поддержка оборудования

Будет ли дистрибутив работать с вашим наличным оборудованием?

**Н**а сервере не место самому современному ПО. Дистрибутивы-ветераны вроде Slackware и Debian неустанно подвергаются программы тестированию. Оборудование, как и ПО, для рабочего сервера рекомендуется не совсем новое, зато поддерживаемое. Выбрав дистрибутив, во избежание любых сюрпризов лучше всего найти «железо», на котором дистрибутив сертифицирован для работы.

RHEL публикует список сертифицированного оборудования. ClearOS и NethServer советуют использовать оборудование

только из этого списка. Тут есть смысл, поскольку эти дистрибутивы основаны на CentOS, созданном на базе кода RHEL. ClearOS также распространяет сервер Clear-BOX с сертифицированным оборудованием. Он доступен в разных спецификациях в зависимости от того, как вы планируете его использовать, количества обслуживаемых пользователей и того, будет ли он использоваться в малом или домашнем офисе или на крупном предприятии. Для домашних и офисных пользователей Koozali рекомендует только три устройства

от HP, и больше ничего. Как и NethServer с ClearOS, для создания списка совместимого оборудования Zentyal использует вышестоящий дистрибутив, Ubuntu. Univention тщательно тестирует оборудование и составляет списки поддерживаемых устройств. Доступный как PDF, список последний раз обновлялся в ноябре 2016 г. и описывает компоненты устройств и совместимую версию Univention. Если ваше оборудование отсутствует в списке поддерживаемого, наилучшей работы вы добьетесь в дистрибутивах на базе RHEL/CentOS.

## Вердикт

- ClearOS ★★★★★
- Univention ★★★★★
- NethServer ★★★★★
- Zentyal ★★★★★
- Koozali ★★★★★

» Koozali надо расширить свой список оборудования, чтобы охватить всё как у RHEL.

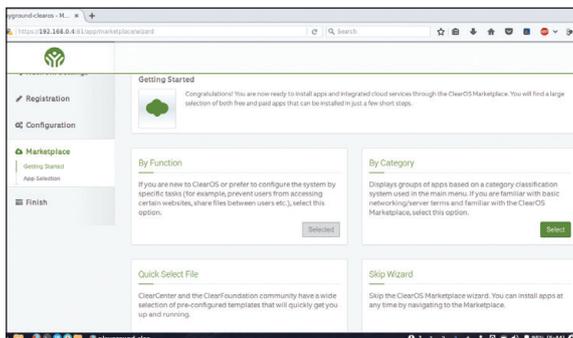
# Дополнительные сервисы

Если у вас есть лишние деньги, потратьте их с пользой.

**В**се эти дистрибутивы могут отследить свои корни до уважаемых дистрибутивов. Лежащая у них в основе система управления пакетами облегчает установку дополнительных пакетов, расширяющих функции этих дистрибутивов за пределы предложения по умолчанию.

Кроме Koozali, который для создания конечного продукта зависит от пожертвованного пользователями, все дистрибутивы в нашем списке предлагают дополнительные сервисы и функции, но небезвозмездно. Эти проекты зарабатывают средства, предлагая дополнительно техническую поддержку, коммерческие приложения и сервисы мониторинга. Некоторые, как ClearOS и Univention, предлагают индивидуальные решения в зависимости от объема развертывания или конкретных требований.

ClearOS предлагает две модели подписки. Базовая, стоимостью £28 в год, дает доступ к 65 приложениям с тестированным качеством, проверенными обновлениями и многим другим. Более дорогая Основная подписка, за £47 в год, помимо базовых



» ClearOS позволяет выбирать программы по функциям или категориям. Второе лучше всего подходит опытным админам, а первое — новичкам в ClearOS.

услуг, предлагает коммерческую защиту от вредоносного ПО и правила антиспама, плюс удаленную резервную копию настройки. Но даже пользователи редакции Сообщества могут приобрести коммерческие приложения, такие как ClearVM Standard, созданный для упрощения использования географически рассеянных серверов и черный список фильтра контента за £78 с Marketplace, и многое еще. Univention также предлагает пакеты коммерческой поддержки от £279 до £1788, которые дают доступ

к получению помощи по электронной почте или телефону и даже индивидуальную настройку. NethServer тоже предлагает автоматические ежедневные обновления, удаленный мониторинг и оповещения в реальном времени, и многое другое в своих 4-х коммерческих вариантах. При приобретении коммерческого варианта, 30 дней доступного бесплатно для ознакомления, Zentyal предоставит техподдержку и обновления программ/безопасности, сверх онлайн-курса для сетевых специалистов.

### Вердикт

ClearOS	★★★★★
NethServer	★★★★★
Univention	★★★★★
Zentyal	★★★★★
Koozali	★★★★★

» Koozali нацелен на создание команды партнеров, которые обеспечат поддержку.

# Простота в работе

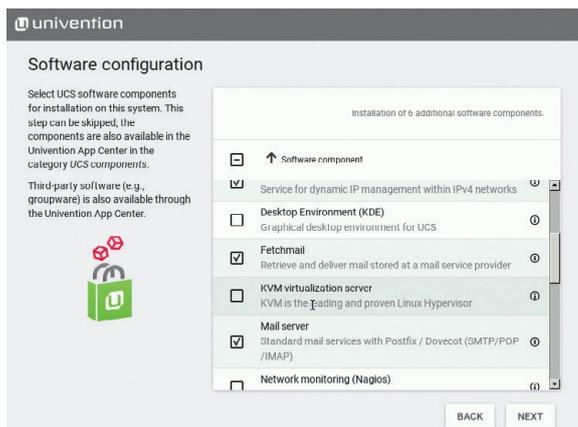
Никому не нравится ломать зубы о гранит науки.

**П**о сравнению с дистрибутивами из недавнего прошлого, установка современного серверного дистрибутива прошла большой путь. Установка типа наведи-и-щелкни, вкупе с простым управлением пакетами, позволяющим установить и настроить такие сервисы, как брандмауэр, DNS сервер и шлюз, делают наши дистрибутивы удобными даже для относительно неопытных пользователей.

Установка сервера Univention быстрая и ровная до определенного момента, но начинает неприятно тормозить, принимаясь за копирование пакетов. Вам придется выбрать, какие серверы устанавливать, например, CUPS, сервер виртуализации KVM, Nagios, почтовый сервер и удобные дополнения, такие как Fetchmail и рабочий стол KDE, помимо прочих. Koozali, по умолчанию использующий программу установки

на curses, во время установки просто летает. Но поскольку он основан на CentOS, Koozali также предлагает на выбор Anaconda, если вам больше нравится установка с помощью мыши. Еще одно производное CentOS, NethServer, тоже предлагает простую установку, но только если на вашем жестком диске уже есть свободное место или отведенный под него раздел. Если нет, то выделение места под NethServer вручную будет подлинным испытанием. К счастью, дистрибутив может запускать автоматическую установку, которая поставит дистрибутив на первый жесткий диск с выбором настроек по умолчанию.

Как и Koozali, Zentyal и ClearOS оставляют большую часть стандартной настройки пост-установочным мастерам. После установки вы должны пройти по всем шагам пост-установочного мастера ClearOS. Сюда входит регистрация в Clear Center, если вы начинающий пользователь, и проверка всех сетевых настроек. Вы не можете пропустить ни одного шага, поэтому сначала вам надо установить все имеющиеся критические обновления для основной системы.



» В отличие от других дистрибутивов, Univention дает выбор сервисов на стадии установки.

### Вердикт

ClearOS	★★★★★
Koozali	★★★★★
Univention	★★★★★
Zentyal	★★★★★
NethServer	★★★★★

» NethServer надо улучшить работу с разделами, для конкурентоспособности.

Бизнес-серверы

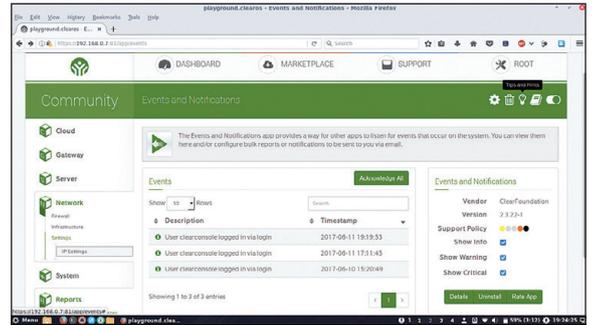
# Вердикт

**В**аш выбор сервера зависит от вашей цели. Единый подход на все случаи не всегда рекомендуется для серверной настройки. Иногда вам нужен всего лишь обычный брандмауэр, и для подобной задачи неразумно настраивать целое серверное решение. Обычно к моменту завершения тестов мы уже можем определить явного победителя, но на сей раз так не вышло. Zentyal сам себя сбросил с пьедестала — поскольку предлагает лишь ограниченное количество сервисов. Но это большая неудача, ведь Zentyal — отлично отлаженный дистрибутив с интуитивным web-интерфейсом, который невероятно облегчает задачу администрирования. Если вам нужен только быстрый в развертывании и настройке сервер шлюза, вам лучше остановить свой выбор на нем.

Univention работает, как и заявлялось, но в нем много странностей, например, учетная запись root с недостаточным уровнем привилегий для выполнения многих функций. Его web-интерфейс также необычен,

и хотя он простой и незахламленный, его не назовешь однозначно интуитивным. Концепция доменов тоже уникальна и требует времени на знакомство с документацией.

Нас впечатлил Koozali. Его можно подогнать для размещения разных серверов и поддержки большого их количества, скажем, файлового сервера, почтового сервера и сервера брандмауэра/шлюза зараз. Но его web-интерфейс оставляет чувство неудовлетворенности и вместе с отсутствием коммерческой поддержки отодвинул дистрибутив на более низкую ступень пьедестала. Зато расширенные функции, такие как удаленный доступ и резервное копирование, предлагаются бесплатно, заставляя другие дистрибутивы опасаться этого молодого проекта.



И у нас остаются два лучших. Очень сложно выбрать между NethServer и ClearOS. Оба базируются на CentOS, что делает их политику релизов и обновлений практически одинаковой. ClearOS предлагает отлаженный целиком и полностью продукт, а NethServer должен усовершенствовать свою установку, уделив повышенное внимание возможностям разбиения на разделы. У ClearOS также немного лучший web-интерфейс; однако даже с учетом данного факта мы советуем пользователям дать шанс обоим проектам, прежде чем выбрать один из них.

» **Первоклассный интерфейс ClearOS должен быть золотым стандартом для всех остальных дистрибутивов.**

**Zentyal — продуманный дистрибутив с интуитивным web-интерфейсом, невероятно облегчает администрирование.**

**I ClearOS** ★★★★★  
 Версия: 7.0 Сайт: [www.clearos.com](http://www.clearos.com) Лицензия: GPL и другие  
 » Всё делает правильно, но это не повод, чтобы почитать на лаврах.

**IV Zentyal** ★★★★★  
 Версия: 5.0 Сайт: [www.zentyal.org](http://www.zentyal.org) Лицензия: Различные FOSS  
 » Ограниченное использование свободной версии его несколько порочит.

**II NethServer** ★★★★★  
 Версия: 7.3 Сайт: [www.nethserver.org](http://www.nethserver.org) Лицензия: GPLv3  
 » Солидный продукт, который лишь самую малость уступает ClearOS.

**V Univention** ★★★★★  
 Версия: 4.2 Сайт: [www.univention.com](http://www.univention.com) Лицензия: AGPL  
 » Свободен для скачивания и использования, но для прочего нужны разрешения.

**III Koozali** ★★★★★  
 Версия: 9.2 Сайт: [www.koozali.org](http://www.koozali.org) Лицензия: GPL  
 » Может достойно соперничать с остальными благодаря отличному интерфейсу.

**Обратная связь**  
 Доверите ли вы свой сервер дистрибутиву с готовыми пакетами, подобному одному из этих? Сообщите нам на [lx.f.letters@futurenet.com](mailto:lx.f.letters@futurenet.com).

## Рассмотрите также...

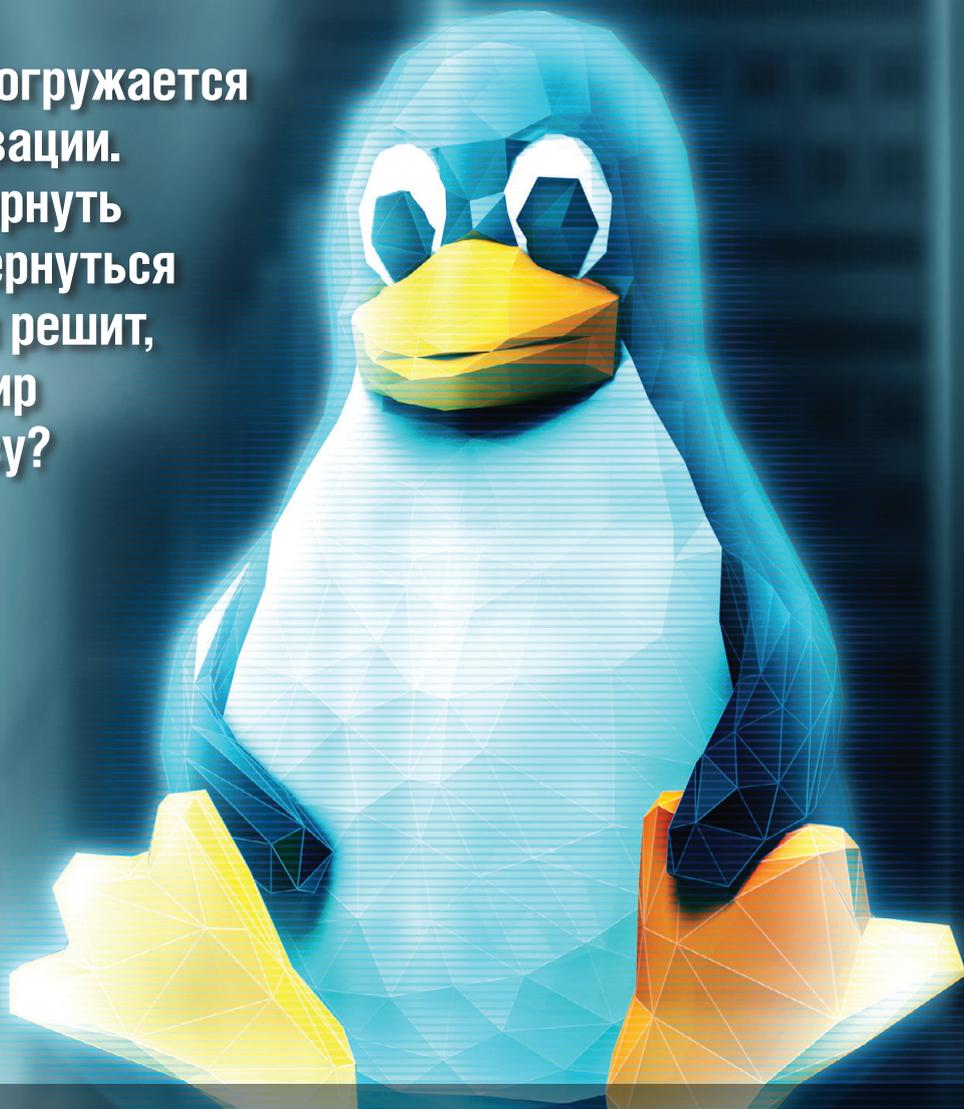
**В**ыбор дистрибутива сервера полностью зависит от планируемого применения. Если вам нужен только сервер с ограниченными или специальными функциями (например, для укрытия вашей конфиденциальности за прокси) либо запуск брандмауэра для защиты, вряд ли будет разумно использовать многоцелевой серверный

дистрибутив. Хотя *iptables* настраивается в большинстве дистрибутивов Linux, спецдистрибутивы вроде IPFire очень легки в развертывании и настройке. Если вам нужно больше функций, например, фильтры спама, управление пропускной способностью, блокировка вирусов и т.п., попробуйте Untangle. Помимо вечно популярного RHEL и множества

его производных от сообщества, серверные редакции Ubuntu и Fedora тоже станут весьма эффективными опциями, если вы знакомы с настольными. А если у вашего дистрибутива нет web-интерфейса администратора, обратитесь к Webmin: это независимое от дистрибутива решение выполнит его задачи, скажем, настроит файлы config. **LXF**

# Виртуализуйте!

**Джонни Бидвелл погружается в пучины виртуализации. Сможет ли он вынырнуть на поверхность и вернуться в реальность — или решит, что виртуальный мир ему больше по вкусу?**



**З**апускать операционную систему на реальном оборудовании давно уж отстой. Тратить долгие часы на настройку оборудования, деликатно прилаживая каждый компонент, чтобы избавиться от его неполадок с помощью правил `udev`, опций `modprobe` и заклинаний вызова демона. Скукота.

Да потом еще последуют долгие часы установки и настройки всех нужных программ, чтобы всё привести в порядок. А не многовато ли тут усилий, чтобы просто, скажем, попробовать дистрибутив Linux? И если что-то пойдет не так, то на вас обрушится лавина лишней работы, поскольку может сгнить ваша обычная установка Linux (такого не должно быть никогда, но зарекаться не стоит).

В сценарии для предприятия надо еще и определиться со стратегией резервного копирования, потому что ошибки и отказ оборудования — вещи вполне возможные. По сути, если вы серьезно относитесь к подобным вещам, вам следует как можно ближе держаться к зеркальному образу компьюте-

он не даром ест хлеб, должен представлять идеальное виртуальное оборудование, которое Linux может быстро и легко понять. Нам всё равно придется настроить программную часть, но как только это будет сделано, мы сможем изменить машину простым нажатием кнопки. Мы можем проверить любые изменения, сделав моментальный снимок машины и внося изменения туда. Если скопированная VM переживет тестирование, то мы будем знать, что и родительская машина тоже его переживет, потому что они идентичны. Просто!

Каким бы образом вы ни применяли Linux, вы всё равно что-нибудь да выгадаете от виртуализации. Поэтому читайте дальше — и узнаете, чем программный компьютер может быть даже лучше реального...

**Как бы вы ни применяли Linux, вы все равно что-нибудь да выгадаете от виртуализации.**

ра, готовому вступить в дело при первом признаке отказа. А поддержка такой штуки бывает делом непростым, даже на идентичном оборудовании.

Сравните это с виртуальной машиной (VM), где не надо переживать по поводу настройки оборудования, поскольку любой гипервизор, если



# Откройте VirtualBox

Ваша первая VM всего в паре щелчков от вас, с любой ОС на ваш вкус.

**Д**ля обычного пользователя, независимо от используемой операционной системы, простейший способ обратиться к виртуализации — *VirtualBox* от Oracle. Она бесплатна, код ее открыт (за исключением некоторых расширений для USB), и у нее есть простой и дружелюбный GUI, позволяющий настроить виртуальную машину всего за несколько щелчков. Хотите ли вы настроить VM Windows XP, чтобы попрактиковаться в навыках хакерства, или попробовать новый дистрибутив Linux (с нашего диска либо тот, что вы скачали сами) — это легко и интуитивно делается в *VirtualBox*.

Чтобы установить *VirtualBox* в Linux, надо всего лишь ввести

```
$ sudo apt install virtualbox
```

или что там требуется в вашем дистрибутиве. Пользователи Windows и macOS [Ред.: — О чем вообще этот журнал?!] могут скачать и установить ее с [www.virtualbox.org](http://www.virtualbox.org); остальные инструкции те же самые.

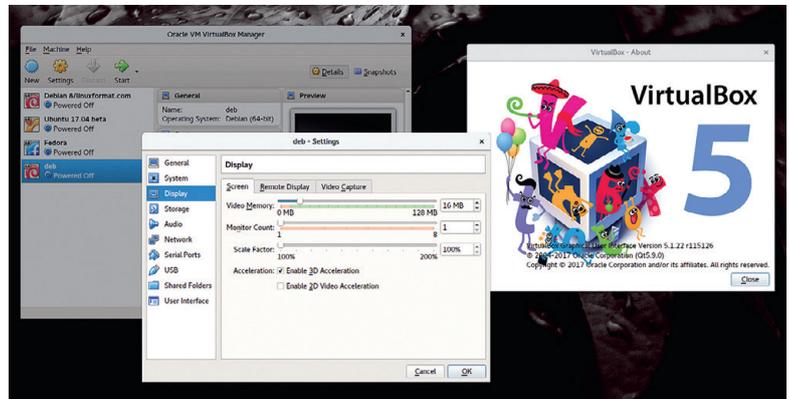
Для создания примера VM мы воспользуемся ISO Debian 9 с нашего диска, но вы можете скачать и использовать любой другой ISO, который окажется у вас под рукой. Наш надо будет скопировать с DVD (он находится в директории **Debian/**) в свою домашнюю директорию, чтобы всё работало быстрее.

## Зарождение VM

Запустите *VirtualBox* и нажмите на кнопку **New** [Новая] слева на панели инструментов. Дайте своей новорожденной VM удобное либо благозвучное имя: если вы назовете ее как-то вроде **Debian-VM**, программа даже поднастроит **Type** и **Version** внизу, а на следующей странице (если вы используете режим мастера **Guided**, а не одностраничный **Expert**) еще и ответит уместный, по ее мнению, объем памяти (для **Debian** — 1 ГБ, но если вы собираетесь активно использовать **Gnome**, понадобится больше). Не забудьте также оставить достаточно памяти для системы хоста.

Следующий шаг — создание виртуального жесткого диска; это файл, который будет содержать все данные о нашей VM. Выберите **Create** [Создать] и укажите образ **VDI** по умолчанию, а на следующем шаге выберите сделать его **Dynamically Allocated** [Отводится динамически]. Это означает, что файл будет расти по мере заполнения виртуального диска — вместо того, чтобы быть большим и сначала практически пустым.

Такая роскошь занимает небольшую плату с производительности, но это нас не очень беспокоит. Простая установка **Debian** на рабочем столе занимает менее 4 ГБ, однако ОС имеют привычку расти



➤ Выделите своей VM больше видеопамати, тогда рабочий стол заработает намного веселее.

со временем, и надо отвести несколько больше места (или заложить некий потенциал на увеличение, если вы используете файл с динамическим ростом), скажем, около 16 ГБ.

Определившись с виртуальным оборудованием, надо выдать машине то, с чего она будет загружаться. Выберите себе машину и нажмите на **Settings** [Настройки] на панели инструментов. Потом перейдите в **Storage** [Хранилище] в левом столбце — и вы увидите, что виртуальный оптический диск (присоединенный к виртуальному контроллеру IDE) настроен так же, как и наш виртуальный жесткий диск. Щелкните по нему, затем нажмите на значок диска справа и выберите файл **Choose Virtual Optical Disk** [Выбрать Виртуальный Оптический Диск] из выпадающего меню. Выберите ISO **Debian**, который мы скопировали в вашу домашнюю директорию ранее. Во вкладке **Screen** в разделе **Settings > Display** установите **Video Memory** на 128 МБ, если ваша видеокарта имеет достаточное **VRAM**. Теперь нажмите **OK** и запустите нашу VM, выбрав ее и нажав на кнопку **Start**.

При желании можно поиграть в **Live Environment**, но всего эффективнее наша VM будет с установленной ОС, так что выберите в меню загрузки опцию **Graphical Installer**. Не беспокойтесь об особой точности деталей установки: если вы вдруг решите, что следовало бы сделать по-другому, можете просто удалить VM и начать сначала. Когда установка закончится, виртуальный диск будет автоматически извлечен. Можете перезагрузить машину и наслаждаться **Debian**.

»

## Будьте, будьте нашим Гостем...

Итак, наша VM работает, но вы, пожалуй, заметите, что работает она вяловато. (Если вам так не показалось, попробуйте воспроизвести видео с YouTube.) Это можно исправить — по крайней мере, отчасти — установив внутри VM *VirtualBox Guest Additions*. Многие дистрибутивы, например, **Ubuntu**, включают их по умолчанию, однако **Debian** к ним не относится, поэтому мы должны скомпилировать их вручную. Не бойтесь, это не сложно. Сначала (в VM) установите компилятор и инструменты модуля с помощью

```
$ sudo apt-get install build-essential module-assistant Dkms
```

Теперь из строки меню в окне VM выберите **Devices > Insert Guest Additions CD Image** [Устройства > Введите образ CD с гостевыми дополнениями]. Вам предложат скачать образ диска; сделайте это (его размер — около 60 МБ). Затем установите образ на виртуальный диск. У диска имеется скрипт **autorun** — правда, по каким-то причинам в **Debian** не работающий; но это не проблема, мы запустим его вручную с помощью

```
$ sudo sh /media/cdrom/VBoxLinuxAdditions.Run
```

Когда работа скрипта успешно завершится, выключите компьютер. После этого мы будем в состоянии включить 3D-ускорение в разделе **Settings > Display**.

Помимо ускорения графики и предоставления возможности динамического изменения размера дисплея, в **Guest Additions** имеются и другие полезные функции: в частности, буфер обмена и папки общего пользования, настраиваемые из меню **Devices**.



# Поговорим с QEmu

VirtualBox — это хорошо, но лучший способ виртуализации — из командной строки с помощью *QEmu*. Готовьтесь к длинным заклинаниям.

## Виртуальное что?

Что делает виртуализация оборудования? Системы AMD-V и VT-x добавили виртуальные режимы MMU, чтобы увеличить скорость операции с памятью, потом — особые инструкции, чтобы гости считали, что работают на встроеном оборудовании, и недавно — виртуальные режимы ввода/вывода для периферийных устройств через AMD-Vi и VT-d.

**М**ы только что увидели, что разобраться с *VirtualBox* легко. Кросс-платформенность делает его удобным для тех, кто знаком с другими ОС. Кроме того, у него открытый код, включая (с 2010 г.) дополнение *Guest*. Однако тот факт, что его выпускает Oracle, кое-кого нервирует, и мы знаем, что многие наши читатели предпочитают работать из командной строки.

*QEmu* (*Quick Emulator*) является просто гипервизором (см. далее), но также эмулятором компьютера и процессора, и мы будем использовать его функции KVM — основанной на ядре виртуальной машины; это встроенный механизм Linux для виртуализации. Он существует с 2007 г., когда распространились процессоры с поддержкой расширений VT-x и AMD-V. KVM обеспечивает интерфейс, который программа (в нашем случае *QEmu*) может применять для размещения нескольких гостевых ОС. KVM позволяет гостям работать со скоростью, близкой к их родной, и поддерживает паравиртуализацию устройств хоста через драйверы *VirtIO*. Она также делает возможной транзитную пересылку PCI с поддерживаемым оборудованием, а значит, попозже мы сможем сделать нечто очень крутое...

Мы настроим простую виртуальную машину с *QEmu*, опять же используя ISO Debian, но вы можете воспользоваться любым, что есть под рукой. Пурристам *QEmu* понравится. Здесь нет GUI и нет возможности сделать настройки VM постоянными: всё приходится

делать через командную строку. К счастью, есть и графические интерфейсы, и одним из них (*virt-manager*) мы воспользуемся — после этого краткого введения в командную строку.

Установите *QEmu* (и 300 МБ всяческих зависимостей) плюс утилиты сетевых мостов (чтобы наша VM могла появиться как законный хост в нашей LAN) с помощью

```
$ sudo apt-get install qemu-kvm bridge-utils
```

Для использования с нашей виртуальной машиной мы создадим «сырой» образ диска объемом 16 Гб. Обеспечьте, чтобы у вас в домашней директории было свободных 16 Гб, иначе это не сработает.

```
$ qemu-img create -f raw lxf-qemu.raw 16GB
```

Вместо *raw* мы можем использовать новый формат, *qcow2*. Он более стойкий и поддерживает встроенные моментальные снимки, но при этом повышает издержки на производительность. Загрузите наш Debian ISO и установите его на наш виртуальный диск, введя

```
$ qemu-system-x86_64 -cdrom /path/to/debian-live-9.0.1-amd64-gnome.iso -boot order=d -drive file=lxf-qemu.raw,format=raw -enable-kvm -m 4G
```

Последний параметр указывает объем памяти. Мы взяли 4 Гб, но вы выделите столько, сколько будет возможно. Объем по умолчанию — 128 Мб, и этого хватит только на самую спартанскую VM. Структура сети по умолчанию (с использованием NAT) мешает программе установки Debian (она не разрешает ping), поэтому список пакетов не обновить прямо сразу. Но продолжить можно и без этого, и попозже мы настроим сетевой мост. По завершении установки мы сможем запустить машину без опций *-cdrom* и *-boot*.

## Эти утомительные строки команд

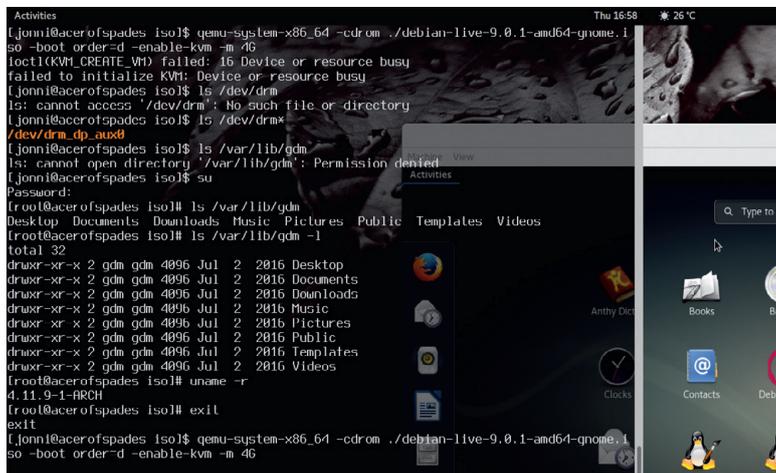
*Virtual Machine Manager* (он же — *virt-manager*) — это инструмент от Red Hat, приносящий *Manager* немного роскоши графического интерфейса в стиле *VirtualBox*. Для установки надо всего лишь ввести

```
$ sudo apt install virt-manager
```

Он обращается к лежащей в основе VM с помощью API *libvirt* (<https://libvirt.org>), которая будет установлена в качестве зависимости. Перед началом использования *virt-manager* надо запустить сервис *libvirtd*, и если вы планируете регулярно его использовать, можете добавить это в автозагрузку системы:

```
$ sudo systemctl start libvirtd
```

```
$ sudo systemctl enable libvirtd
```



➤ На свете много опций командной строки *QEmu*, Горацио...

## Проблемы оборудования

Виртуализация существует уже давно, и на любой старой машине, способной запустить настольный Linux, может работать гостевая VM. Однако если вы хотите, чтобы эта VM справлялась со сложными задачами, нужно нечто более современное. В частности, процессор, поддерживающий расширения виртуализации, Intel VT-x или AMD-V, которые имеются на большинстве 64-битных CPU настольных ПК с 2006 г. Вы можете это выяснить, запустив `egrep '(vmx|svm)' /proc/cpuinfo`

Если расширений там нет, то команда ничего не вернет.

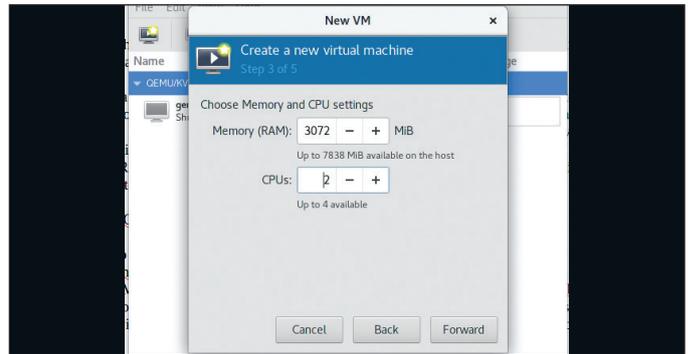
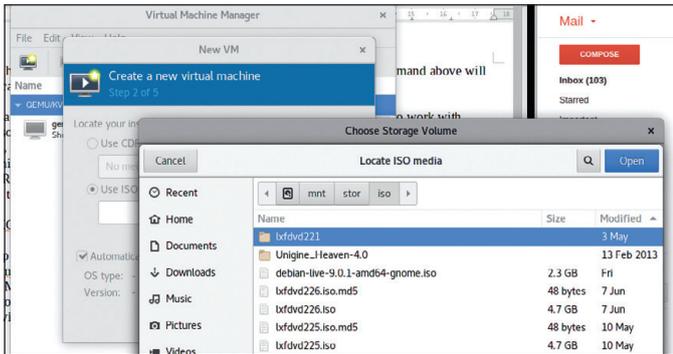
Учите, что всё это можно отключить по умолчанию в BIOS/UEFI, где у них часто разные названия. Так что проверьте — команда `egrep` выше всё равно покажет возможности, даже если они отключены.

Виртуальные машины откусывают много памяти, а работа как с гостем, так и с хостом становится куда менее приятной, если этот ресурс находится под угрозой. Небольшой виртуальный

сервер, на котором не запущен пышный GUI, может спокойно обойтись 512 Мб; а вот запуск рабочего стола потребует столько же ресурсов, сколько потребовалось бы физически.

В результате компьютер с 4 Гб ОЗУ легко изнурить одной-единственной VM, которая не очень-то и загружена. А ведь мы живем в эпоху, когда браузеры радостно перемалывают гигабайты памяти, просто чтобы воспроизвести простейшие видео с котятками.

## Запустите виртуальную машину

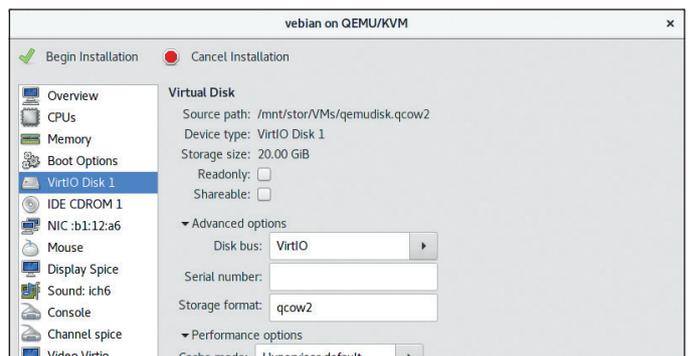
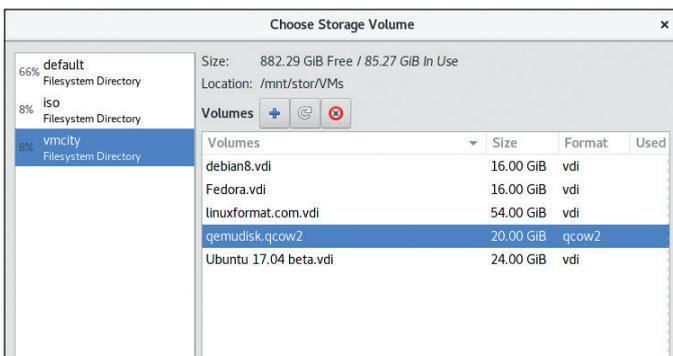


### 1 Запуск virt-manager

Запустите *Virtual Machine Manager* (для соединения с *libvirt* вам, возможно, придется ввести пароль) и нажмите на кнопку *New* на панели инструментов. Выберите *Local install media > Use ISO Image > Browse > Browse Local*, чтобы перейти к требуемому вам ISO. Мы уже дважды упоминали ISO Debian, поэтому давайте придерживаться его.

### 2 Распределение ресурсов

Выделите своей машине достаточно памяти, и если у вас несколько ядер, подумайте о создании более одной VM. Предоставьте своей машине виртуальный образ диска. Первая опция создаст образ *qcow2*, вторая позволит вам указать тип образа и выбрать альтернативное местоположение (по умолчанию это */var/lib/libvirt/images*).

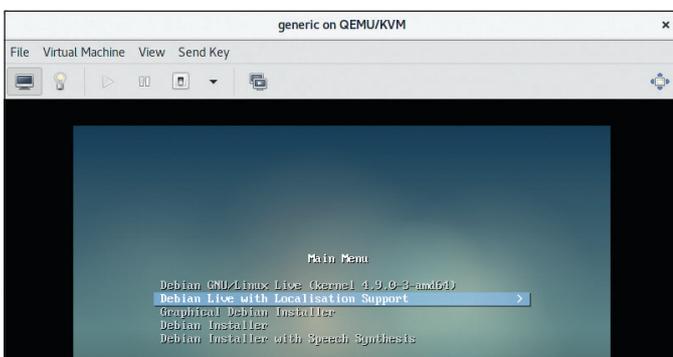


### 3 Имена и сети

Дайте вашей машине имя — значимое или симпатичное (или сочетающее эти свойства) — и отметьте окно *Customise configuration before install* [Индивидуальная настройка перед установкой]. Если вы установили *bridge-utils* и хотите, чтобы ваша машина была доступна в LAN, измените сетевое устройство на *macvtap* в соответствии с вашей сетевой картой и в качестве *Source* выберите *Bridge*.

### 4 Выход на скоростную полосу

Вместо эмуляции оборудования мы можем его паравиртуализовать с помощью драйверов *VirtIO*. Если вы настраиваете гостя *Windows*, то надо добавить драйверы во время установки (см. раздел *Транзитная пересылка PCI*, далее), но с гостями *Linux* всё в порядке. Выберите *IDE Disk 1* и смените его шину на *VirtIO*. Опционально, сделайте это для устройства *QXL* и затем включите *3D-ускорение*.



### 5 Загрузите VM

Выберите *Begin Installation* [Начать Установку] вверху слева. Если будут ошибки, попробуйте отменить всё, что касалось *VirtIO* выше. Обратите внимание, что окно *QEmu* захватит курсор мыши. Не бойтесь, его можно освободить, нажав правый *Ctrl* и правый *Alt*. Загрузившись в среду *live* и запустив *ip a*, вы увидите, что у вашей VM есть IP-адрес в вашей подсети LAN.

### 6 Установка и перезагрузка

Как и раньше, установите гостевую ОС. Если на шаге 3 вы выбрали сетевой мост (*Bridge*), то программа установки *Debian* не запустится при поиске зеркала. Перезагрузитесь и наслаждайтесь вашей VM. Вы можете развернуть VM на весь экран с помощью значка вверху справа. Если вы освободите курсор мыши и наведете его на центральную часть вверху, меню позволит вам выйти.

# Контейнеры и Docker

Виртуализация — не единственная цель в стремлении к изоляции...

**П**ри всех своих достоинствах, каждая новая раскрутка VM означает новую копию операционной системы. Это означает немалое количество скопированных данных и потраченных усилий. Виртуальная ОС всё равно потребует времени на установку, и несмотря даже на то, что этот процесс отчасти автоматизируется, всё равно большинству VM требуется настройка, прежде чем вы получите от них нечто путное.

Новые прорывы в технологии ядра (cgroups и namespaces) открыли новый, более прямой способ изоляции приложений и сервисов без установки всей новой ОС. Такие «контейнеры» уходят корнями в «узилища [jail]» и *chroots* BSD, т.е. сама по себе эта идея (некий доступ к ядру хоста при ограничениях в пространстве пользователя) не нова; но только с появлением Linux Containers (LXC) в 2008 г. стало возможно делать это с достаточной уверенностью.

Эта идея реализовалась в 2013 г., когда появился *Docker*. Контейнеры позволяют развертывать одинаковые приложения на любом дистрибутиве. Любому системному администратору, которому приходилось переносить старые web-приложения в новую инфраструктуру, понятно, от скольких мучений это избавляет.

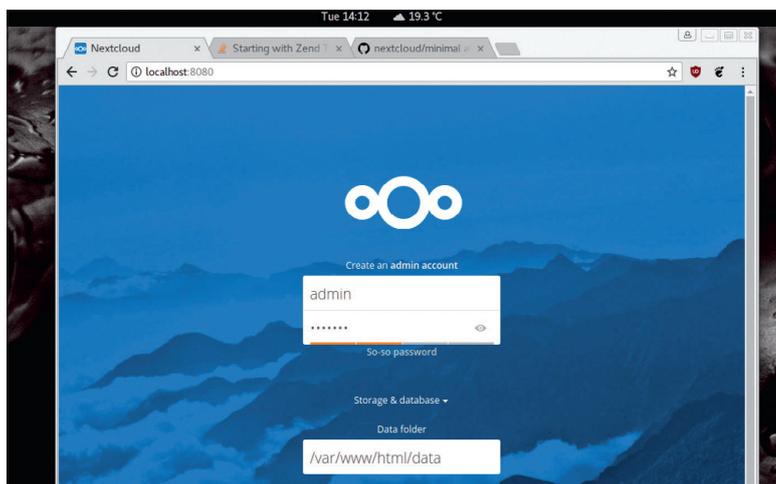
## К счастью, все это легко настроить с помощью магии операторов командной строки.

Благодаря Docker Hub теперь можно скачать образ сервера электронной почты, экземпляра *NextCloud*, комплекса NAS или сервера *MPD* и запустить за считанные секунды. Конечно, ни одно (нетривиальное) приложение нельзя изолировать, как остров: ему может потребоваться доступ к сети хоста, аудио или частям файловой системы. Ему даже может потребоваться доступ к другому контейнеру Docker. К счастью, всё это легко настроить с помощью магии операторов командной строки.

### Китовый труд

Для нашей первой вылазки в волшебный мир контейнеров мы настроим контейнер *Nextcloud*, взяв его официальный образ. Первый шаг — установить *Docker* и инструмент *docker-compose*, что сведется к

➤ Готовые к использованию образы *NextCloud* и *MariaDB* означают, что вы можете настроить собственное облако [Ред.: — Так быстро?] за считанные минуты.



```
$ sudo apt install docker docker-compose
```

или эквиваленту. Пользователям дистрибутивов, не являющихся производными от Debian, будет приятно узнать, что все остальные команды в этом разделе не зависят от дистрибутива (кроме употребления *su* вместо *sudo*, в зависимости от того, как вам или вашему дистрибутиву больше нравится работать). Скачайте образ *Docker* и запустите демон, с помощью

```
$ sudo docker pull nextcloud
$ sudo systemctl start docker
```

Он отберет 700 МБ — это немало, и помните, что приложение *NextCloud* занимает 100 МБ плюс у нас есть *Apache*, PHP 7 и ряд расширений, да еще база данных *SQLite*. Если вы планируете позволить нескольким людям (скажем, более пяти человек) использовать этот экземпляр *NextCloud*, то вам нужно использовать соответствующую базу данных SQL, такую как *MariaDB*. Вы можете организовать ее в другом контейнере, как описано в официальном Readme (см. <https://github.com/nextcloud/docker>), но мы пока будем использовать *SQLite*.

Запустите контейнер:

```
$ sudo docker run -d -name lxfcloud -p 8080:80 nextcloud
```

Опция *-d* велит *Docker* работать в фоне (отдельно), а часть *-p* переадресует порт 80 на контейнере на порт 8080 на нашем хосте. Часть *-name* является опциональной; если вы ее не укажете, она будет присвоена случайным образом. Если эта команда не выдаст вам никаких ошибок, запустите браузер и перейдите на <http://localhost:8080>. Любуйтесь! Полностью функционирующая установка *NextCloud* без всякой возни с файлами настройки *Apache* или разрешениями директории; она просто работает. Всё, что вам надо сделать — ввести аутентификационную информацию администратора и нажать *Finish Setup*.

При остановке контейнера командой

```
$ sudo docker stop lxfcloud
```

все изменения сохраняются, если не обновляется лежащий в основе образ — см. далее. Поэтому мы можем возобновить его с помощью

```
$ sudo docker start lxfcloud
```

Обратите внимание, что мы используем имя нашего контейнера, а не имя образа (*nextcloud*), поскольку последний запустит новый, ненастроенный экземпляр *NextCloud*.

Случайные контейнеры, заполняющие директорию */var/lib/docker*, могут потратить место на диске, поэтому здесь очень важно выполнять «уборку». Остановленные контейнеры не показываются в выводе *docker ps*, но вы можете увидеть, скажем, пять последних остановленных контейнеров, запустив

```
$ sudo docker ps -n 5
```

Затем удалите всё, что больше не требуется, с помощью *docker rm <имя или id контейнера>*. Ненужные образы (предварительные сборки, скачанные по *docker pull*) можно удалить командой *docker rmi <имя образа>*. Нашей следующей хитростью будет настройка еще более функционального производимого экземпляра *NextCloud*, поэтому перед началом работы остановите и удалите все работающие контейнеры. Мы воспользуемся инструментом *docker-compose*, он упростит нам жизнь. Нам понадобится официальный образ *MariaDB*, который мы найдем командой

```
$ sudo docker pull mariadb
```

Мы соединим образы *nextcloud* и *mariadb*, чтобы они могли общаться друг с другом, и у нас будет смонтирован том хранилища на нашем хосте, чтобы мы могли легко получить доступ к данным

## Выход за рамки Docker

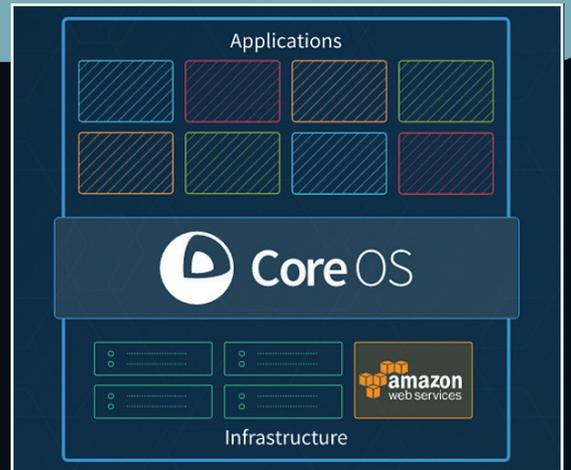
Docker — не единственный плод контейнеров, и когда говорят «контейнеры Docker», то подразумевается, что их (файлы, хранящиеся на диске и их метаданные) можно прекрасно перенести в другие среды исполнения.

Двумя примерами будут rkt и runc в CoreOS, еще одним — libvirt-lxc. В Systemd имеется легковесный механизм контейнеров, systemd-nspawn. Отличное свидетельство его использования для запуска Steam и предотвращения загрязнения вашей системы древней 32-битной средой исполнения есть на <http://ludiclinux.com/Nspawn-Steam-Container>.

Стандарты Snap и Flatpak для не зависящих от дистрибутива приложений активно

используют технологию контейнеров, обеспечивая функции изоляции. Хотя многие приложения, распределяемые таким образом, не работают так же, как их аналоги (кхе-кхе, LibreOffice), происходит тихая революция, которая уменьшит головную боль разработчиков и даст пользователям шанс получать новые релизы скорее.

Container Linux от CoreOS, бывший CoreOS Linux — это дистрибутив, созданный для облака, в котором всё устанавливается в своем контейнере. Джесси Фрейзелл [Jessie Frazelle] из Google создала сильно измененную версию рабочего стола Container Linux, в которой всё работает в контейнерах без root: <https://blog.jessfraz.com/post/ultimate-linux-on-the-desktop>.



» CoreOS Container Linux — это, возможно, будущее облачных дистрибутивов Linux. Можно запускать и дома.

NextCloud. Всё это определяется в файле **YAML**, который вам надо транскрибировать (либо скопировать и вставить с Docker GitHub). Неплохо также изменить secretpassword на ваш собственный секретный пароль:

```
version: '2'
volumes:
  nextcloud:
  db:
services:
  db:
    image: mariadb
    restart: always
    volumes:
      - db:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=secretpassword
      - MYSQL_PASSWORD=secretpassword
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud
  app:
    image: nextcloud
    ports:
      - 8080:80
    links:
      - db
    volumes:
      - nextcloud:/var/www/html
    restart: always
```

Сохраните это как **docker-compose.yml** (использование имени файла по умолчанию сэкономит в дальнейшем нажатия на клавиши), и если вы желаете быть аккуратным, поместите в директорию, скажем, **~/docker-nextcloud/**. Затем войдите в эту директорию по cd и запустите

```
$ sudo docker-compose up -d
```

Если всё прошло хорошо, вы сможете снова перейти на <http://localhost:8080> для настройки. А если не сработает, проверьте логи на предмет ошибок с помощью **docker-compose logs**. Запуск **docker ps** должен показать два запущенных контейнера (или больше, если вы не прибрались за собой), с именами **docker\_nextcloud\_1** и **docker\_db\_1**. Контейнер базы данных доступен из контейнера Nextcloud через имя хоста **db**, поэтому когда вы выбираете MySQL/MariaDB в настройках Storage & database [Хранение и базы данных] на странице настройки, введите его в поле хоста базы данных. Вам надо будет ввести аутентификационную информацию для еще одной

учетной записи администратора, а также ввести nextcloud в полях имени базы данных и имени пользователя, и ваш секретный пароль, заданный выше, в окне пароля. На настройку таблиц и столбцов уйдет несколько секунд, так что не волнуйтесь, если после нажатия на кнопку Finish Setup всё замрет.

Если нам надо остановить наш слаженный дуэт контейнеров, достаточно всего лишь запустить **docker-compose stop** из директории, содержащей наш файл **YAML**. А перезапустить, как вы и понадеялись, с помощью **docker-compose start**. Чтобы полностью стереть нашу композицию, используйте **docker-compose down**. Указав два разных тома во всё том же **YAML**, мы обеспечим сохранение данных при обновлениях (которые можно настроить с помощью **docker-compose up --upgrade**). Тома хранятся отдельно, и мы можем выяснить, где именно, изучив (довольно объемный) вывод от, например, **docker inspect docker\_db\_1**:

```
...
"Mounts": [
  {
    "Type": "volume",
    "Name": "docker_db",
    "Source": "/var/lib/docker/volumes/docker_db_data",
    "Destination": "/var/lib/mysql",
    "Driver": "local",
    "Mode": "rw",
    "RW": true,
    "Propagation": ""
  }
]
...
```

## Нет HTTPS — нет сервиса

Мы уже видели, насколько Docker упрощает развертывание сервисов, и мы настроили экземпляр NextCloud, который (в зависимости от вашего оборудования и сетевого соединения) должен поддерживать несколько пользователей. В вашей LAN он вполне годится, однако ни при каких обстоятельствах в его настоящем виде он не должен быть доступен через Интернет.

Мы не настроили на web-сервере HTTPS, и у нас нет места для этого. Если бы оно было, то вместо изменения существующего образа NextCloud/Apache, мы бы добавили в смесь еще один контейнер, на сей раз — обратный прокси Nginx. Пример настройки имеется на <https://github.com/JrCs/docker-letsencrypt-nginx-proxy-companion>, и он, в свою очередь, требует образа nginx-проxy. Это будет отличный проект для дождливых выходных, так что расскажите нам, что у вас получилось.

# Пересылка графики

Обретите святой Грааль игр Linux — отсутствие необходимости загружать Windows ради игр — благодаря транзитной пересылке PCI и OVMF.

**И**гры Linux прошли долгий путь за относительно короткое время. Причиной тому разные факторы: развитие индустрии, поддержка Linux основными игровыми движками, Valve и их SteamOS, улучшения драйверов отображения и порты от Feral и Virtual Programming. И всё же есть игры, которые никогда не придут в Linux, и игры, которые работают там хуже.

Серьезным игрокам в некоторых ситуациях придется обратиться к ОС из Редмонда, и это плохо. Windows уже давно можно

запускать на виртуальной машине, но даже на быстром оборудовании это не очень хорошо работает. 3D-ускорение в *VirtualBox* мало что дает для DirectX 10, 11 или 12. Есть также *Wine* — чудесная программа, позволяющая некоторым играм прекрасно работать на Linux; однако на каждую работающую игру приходится три, которые выдают ужасные сбои.

Однако в последние годы новый замечательный акроним VFIO (Virtual Function I/O) позволяет устройствам PCI перераспределяться, перенастраиваться и в принципе работать с доступом из ядра Linux. В результате можно передавать оборудование прямо в виртуальную машину, способную взаимодействовать с ним почти что с изначальной скоростью. Если на этой виртуальной машине работает Windows, и оборудование, которое вы передаете ей, является достаточно мощной видеокартой, то — при некоторых условиях — вы сможете играть во все те игры, которые не работают (хорошо) на Linux.

Чтобы это сработало, вам нужны два GPU. Есть также ряд других требований, описанных во врезке внизу. Были сообщения о разного рода ситуациях, когда это не работает, поэтому мы будем благодарны за любую обратную связь. Хотим сообщить, что у нас был успешный опыт с материнской платой Gigabyte 990FX-UD5 (non-UEFI) на процессоре Athlon 8350, Nvidia GTX 960 (одной из лучших видеокарт 2012 г.) в качестве хоста, и Radeon 290 OCX в качестве гостя. Все они потребляют весьма много энергии, но 1000-В блок питания [Ред.: — Легкий перебор] обеспечит нормальное питание всех компонентов.

## IO(MMU)

После включения IOMMU (см. врезку) в BIOS вам понадобится добавить в системный загрузчик опцию ядра: либо `intel_iommu=on`, либо `amd_iommu=on`. Вы можете сделать это, отредактировав `grub.cfg` напрямую, или же посредством рекомендованного метода ее добавления в строку `GRUB_CMDLINE_LINUX_DEFAULT` в `/etc/default/grub` и запуска `grub-mkconfig`. Теперь перезагрузитесь, сохранитесь и запустите такой скрипт:

```
#!/bin/bash
shopt -s nullglob
for d in /sys/kernel/iommu_groups/*/devices/*; do
    n=${d##*/iommu_groups/*}; n=${n%/*}

```



➤ Переходим границы: мы использовали Nvidia GTX 960 в качестве хоста и немного более новый и быстрый Radeon 290 OCX в качестве гостя.

## Аппаратные требования

Чтобы этот трюк сработал, требуется достаточно новое оборудование, но и с современным оборудованием проблемы не исключены.

Для хоста сгодится любая видеокарта, даже встроенная. Если у вас два отдельных GPU, надо ознакомиться с комбинацией клавиш для входа в BIOS/UEFI, поскольку, по закону Мэрфи, ваш компьютер попытается загрузиться не с той видеокарты. Видеокарта гостя должна поддерживать UEFI (а материнская плата не должна), поскольку мы загружаемся с OVMF. Возможно, обновленная BIOS GPU доступна от производителя (некоторые сообщали, что им удалось ее получить просто в ответ на вопрос),

и вы можете передать ее в *QEmu* через опцию `-romfile` вместо использования вашей видеокарты.

В принципе, подойдет всё, что произвели после 2012 г. Ваша материнская плата должна поддерживать IOMMU (оно же — Directed I/O), который, возможно, придется включить в BIOS, и ваш CPU тоже должен его поддерживать (Intel называет это VT-d, а AMD — AMD-Vi; учтите, что всё это в дополнение к более общим VT-x и AMD-V, упомянутым ранее).

Кроме того, возможно, вы захотите потратиться на переключатель KVM (ничего общего с ядром) или на дополнительную клавиатуру/мышь/тачпад, поскольку (вне подключения VNC или SPICE)

машина-гость не будет иметь доступа к устройствам ввода хоста. Вы можете сделать передачу через устройства USB, но если у гостя что-то пойдет не так, они не будут освобождены, а значит, вы не сможете контролировать хост.

Если вам не нравится путаться в проводах, то вам также понадобится монитор с несколькими входами (и еще один видеокабель) или второй монитор. Если вы используете карту Nvidia, вы можете столкнуться с жуткой “error 43”, которую можно обойти, поднастроив вашу *QEmu*. Если у вас система Ryzen, то обновление BIOS может исправить хитрые группировки IOMMU.

## Установка Windows

Не ждали такого в *Linux Format*, а? Это практически то же самое, что описано в статье ранее, разве что вам понадобится ISO Windows, который можно скачать с Microsoft и бесплатно использовать 30 дней.

В разделе *Overview* измените BIOS на UEFI. Возможно, вам понадобится перезапустить *libvirt*, чтобы эта опция стала доступна. В разделе *CPU* надо изменить модель на *host-passthrough*; это может не появиться в выпадающем меню, тогда понадобится ручной ввод. Вместо виртуального диска IDE по умолчанию рекомендуется использовать контроллер *VirtIO SCSI* и виртуальный диск *SCSI*, но Windows нужны дополнительные драйверы, прежде чем программа установки его распознает. Драйверы можно найти на <https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/latest-virtio>, а ISO смонтирован внутри VM как устройство *CD-ROM*.

Мы провели установку до того, как приступить к колдовству транзитной передачи. По завершении выключите VM.

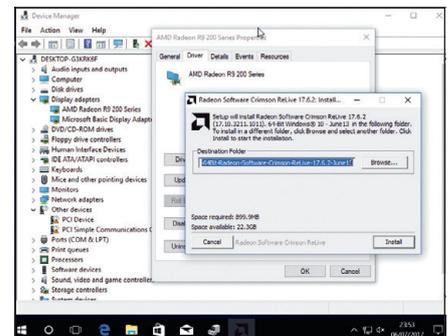
Виртуальные устройства (канал *Spice*, видеоадаптер *QXL* и эмулированные периферийные

устройства) будут нас только тормозить, так что удалите их. Это, конечно, означает, что единственная возможность увидеть происходящее в VM — через переданный GPU. Поэтому нам бы лучше с этим разобраться. Нажмите *Add Hardware* и выберите нужные *PCI Host Devices* (все в группе *IOMMU*). Вам нужна передача клавиатуры *USB* и устройства мыши, но надо также сохранить одну для управления хостом (см. врезку); для этого хорошо подходят дешевые беспроводные клавиатуры/тачпады.

Будем надеяться, что когда вы загрузите VM, запустится транзитная передача GPU (возможно, придется разбудить дисплей через клавиатуру); если нет — пока не отчаивайтесь. Если всё отлично, двигайтесь вперед и устанавливайте драйверы Windows; далее исследуйте раскладку CPU и прочие тонкости производительности.

На эту тему можно сказать куда больше, и множество материалов для чтения вы найдете в Сети. Самый полный ресурс — страница Arch Wiki на [https://wiki.archlinux.org/index.php/PCI\\_passthrough\\_via\\_OVMF](https://wiki.archlinux.org/index.php/PCI_passthrough_via_OVMF); она охватывает массу

пограничных случаев и в большей или меньшей степени применима и к другим дистрибутивам. Есть также несколько обучающих клипов на YouTube, если визуальное представление материала вам больше подходит.



Так можно прекрасно поиграть, не загружая Windows, но в долгосрочной перспективе всё равно понадобится лицензия на Windows.

```
printf 'IOMMU Group %s' "$n"
lspci -nns "${d##*/}"
done;
```

Образуются группы *IOMMU* для вашего оборудования. Устройство может быть передано в VM только вместе с другими членами ее группы *IOMMU*. В нашем случае мы увидели:

```
...
IOMMU Group 16 01:00.0 VGA ... Hawaii XT / Grenada XT
[Radeon R9 290X/390X] [1002:67b0]
IOMMU Group 16 01:00.1 Audio device ... Hawaii HDMI
Audio [Radeon R9 290/290X / 390/390X] [1002:aac8]
...
```

Итак, нам придется сделать транзитную передачу *HDMI Audio*, а также *GPU*. Здесь мы не касаемся аудио, но заставить его работать несложно посредством пропускания аудио устройства или использования сетевых возможностей *PulseAudio*. Посмотрите, как сгруппированы ваши *USB*-контроллеры, чтобы вы могли отвести один на передачу, не затрагивая другие устройства. Возможно, вам придется переставить вашу видеокарту в другой слот *PCIe*, если она сгруппирована с важными элементами: некоторые слоты относятся к *CPU*, а некоторые — к шине, и обращаться с ними надо по-разному.

Для передачи *GPU* надо проверить, что к нему не привязаны драйверы ядра. Это означает, что его состояние позволит VM делать с ним то, что она сочтет нужным. Современный способ сделать это — с помощью драйвера *vfio-pci* и пар производитель — ID устройства для оборудования, которое надо изолировать. Если вы используете одинаковые видеокарты для хоста и гостя, такое не срывает, поскольку их ID будут одинаковы. Вы найдете информацию о том, как это обойти, по ссылкам в конце статьи. В нашем случае мы передаем *GPU* и устройства *Audio* выше, поэтому создаем файл `/etc/modprobe.d/vfio.conf`, содержащий

```
options vfio-pci ids=1002:67b0,1002:aac8
```

Чтобы обеспечить загрузку *vfio-pci* перед графикой, надо убедиться, что модули *vfio* являются частью *initramfs*. Поэтому отредактируйте `/etc/initramfs-tools/modules` (в Arch Linux — `/etc/mkinitcpio.conf`) и убедитесь, что перед каждым видеодрайвером появляются четыре модуля: *vfio*, *vfio\_iommu\_type1*, *vfio\_pci* и *vfio\_virqfd*. Затем вам понадобится восстановить *initramfs* с помощью

`sudo update-initramfs -u`. Теперь перезагрузитесь и проверьте вывод `dmesg | grep -i vfio`. Мы с облегчением увидели

```
[ 1.488517] VFIO — User Level meta-driver version: 0.3
[ 1.489469] vfio-pci 0000:01:00.0: vgaarb: changed VGA decodes:
olddecodes=io+mem,decodes=io+mem:owns=none
[ 1.504260] vfio_pci: add [1002:67b0[ffff:ffff]] class
0x000000/00000000
[ 1.520925] vfio_pci: add [1002:aac8[ffff:ffff]] class
0x000000/00000000
[ 5.429259] vfio-pci 0000:01:00.0: vgaarb: changed VGA decodes:
olddecodes=io+mem,decodes=io+mem:owns=none
```

— это неопровержимо доказывает, что *vfio-pci* полагается на принадлежащее ему по праву. Если ваш вывод не такой же, то, возможно, драйвер слишком скромный. Вместо этого проверьте вывод `lspci -nkn -d 1002:67b0`, изменив соответствующим образом ID. Это

## VFIO позволяет устройствам PCI работать с доступом из ядра Linux.

покажет, какой именно драйвер используется — неважно, что загрузился конкурирующий драйвер (например, *pouveau*, *nvidia*, *radeon*), был бы он привязан к устройству.

*Open Virtual Machine Firmware (OVMF)* — это свободный порт прошивки *Tianocore UEFI* от Intel для *QEMU*. Установите его с помощью `sudo apt-get install ovmf`. Мы предполагаем, что у вас уже установлены *libvirt*, *virt-manager* и *qemu*. Но надо сообщить *libvirt*, где найти бинарники *OVMF*, что включает замену строки `nvram =` в `/etc/libvirt/qemu.conf` на

```
nvram = [
    "/usr/share/OVMF/OVMF_CODE.fd:/usr/share/OVMF/OVMF_VARS.
fd"
]
```

В Ubuntu могут возникнуть проблемы с разрешениями (см. <http://blog.system76.com/post/139138591598/howto-qemu-w-ubuntu-xenial-host-uefiguest>), Arch использует другие имена файлов, но у нас работал нормально.

# От физического к виртуальному

Подумайте о тестовой площадке на виртуальной версии вашего рабочего стола или сервера — или просто выместите на этой версии злость!

**П**реобразование из физического в виртуальное (P2V) включает создание виртуальной версии физической системы. Если физическая система достаточно проста — скажем, сервер, на котором только что установлен Ubuntu Server — то будет намного проще создать новую VM и повторить установку, чем всё переносить.

С другой стороны, если на этой машине работают самые разные сервисы, то дублирование и настройка всего будет довольно сложной и окажется напрасными усилиями. Однако перенос всего в виртуальное пространство тоже имеет свои подводные камни. Во-первых, любой файл конфигурации, относящийся к оборудованию, будь то сетевые интерфейсы, жесткие диски (или UUID

их разделов) или видеоадаптеры, требует настройки, потому что эти устройства не существуют на целевой VM. У физической машины может быть своеобразная настройка LVM и/или RAID, трудно воссоздаваемая на VM, но ее ценность будет невелика, поскольку добавление и резервное копирование виртуальных жестких дисков — дело нехитрое. И наоборот, если у вас несколько разделов и копия всего дерева root (минус псевдофайловые системы, вроде `/dev` и `/proc`), то изменения должны быть внесены, ибо теперь у вас только один раздел.

## Потертыые данные

Есть инструменты, которые могут автоматизировать процесс P2V (например, *VMware vCenter Converter*), но всё это можно сделать и из удобной среды командной строки. Мы начнем с создания «сырого» образа диска и заполнения его данными, а затем конвертируем этот образ в то, что может быть использовано *VirtualBox* или *QEmu*, по вашему усмотрению.

Мы исключим из процесса копирования все ненужные файлы или директории, однако у вас должна быть разумная оценка о требуемом свободном пространстве. Нужно также, чтобы это пространство было доступно для хранения образа. Скажем, мы уверены, что сможем втиснуть всё в 16 Гб. Тогда создадим наш образ так:

```
$ falldate -l 16GiB ~/p2v.img
```

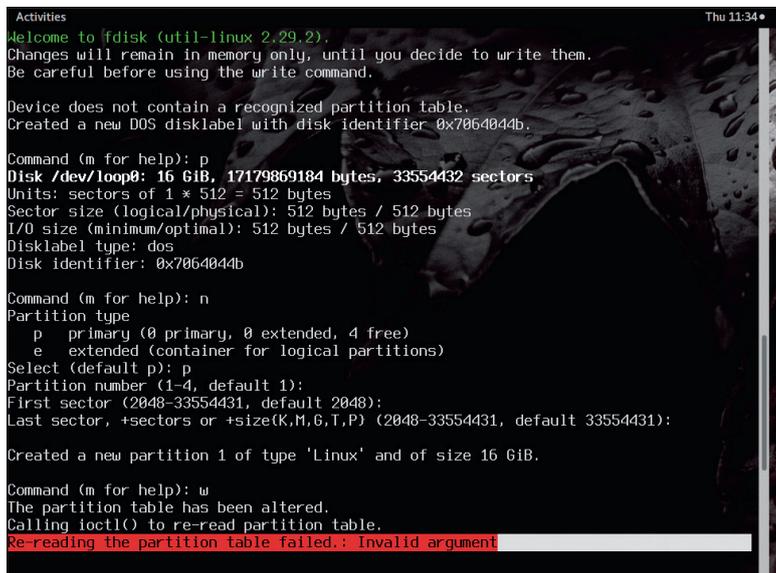
Теперь загрузим модуль обратной петли (это часть новых ядер Ubuntu, так что там он будет не нужен) с помощью `modprobe loop` и подключим образ с помощью

```
$ sudo losetup /dev/loop0 p2v.img
```

Необходимо разбить это устройство на разделы — для простоты мы ограничимся одним разделом. Запустите *fdisk* (или другую любимую вами утилиту разбиения на разделы):

```
$ sudo fdisk /dev/loop0
```

Затем создайте новый первичный раздел, занимающий весь диск, введя `n` и `p`, затем примените предложение по умолчанию для начального и конечного секторов. Запишите изменения и выйдите, введя `w`. Теперь там должен быть дополнительный узел



➤ Запустите `partx -u /dev/loop0`, если после разбиения файла образа у вас появилось это сообщение. Устройства обратной петли довольно причудливы.

## Прошлое и будущее виртуальных машин

К концу прошлого десятилетия наметилась повсеместная тенденция виртуализовать всё. Люди осознали, что можно получить потенциальную экономию, нашпиговав компьютер кучей VM вместо традиционного подхода «один сервер — одна задача».

Отчасти это разумно: почтовый сервер большую часть времени может простаивать, поэтому, возможно, ресурсов хватит на какой-нибудь движок базы данных на том же компьютере? Однако запускать оба сервиса на одной и той же ОС рискованно, поскольку сбой одного сервиса может привести к отказу всей системы, или, возможно, мы не можем доверить ОС работу с ресурсами, если оба сервиса вдруг окажутся очень заняты.

Поэтому мы виртуализуем оба сервиса, и благодаря развитию оборудования ни один из них

не пострадает от потери производительности, и оба будут изолированы друг от друга. К сожалению, эту идею легко довести до крайности и начать виртуализовать всё подряд, втискивая то, что должны выполнять пять или шесть компьютеров, в один.

### Истощение ресурсов

Это заметно повышает риск перегрузки системы и означает также, что из-за сбоя оборудования могут отказать пять или шесть сервисов. Критически важные сервисы всё же достойны отдельной машины; какими бы ни были причины для их виртуализации, объединение их с другими ресурсоемкими машинами приведет к перегрузке. Кроме того, нравится вам это или нет, но виртуализация чего бы то ни было приводит к появлению дополнительного уровня сложности, причем в той области, которая

и без того сложна (см. интервью **LXF226**). Вполне понятно, что некоторые программы плохо отреагируют на свою виртуализацию, а диагностика всего этого бывает весьма затратна по времени, стоимости и эмоциям.

Примерно в 2014 г. мы наблюдали такой же ажиотаж по поводу контейнеров. Бессмысленные споры о том, что контейнеры лучше VM, кипели повсеместно, а между тем эти две технологии предназначены для решения разных задач. Если вы попытаетесь использовать контейнеры как VM, то VM выиграют, а если вы настроите VM на запуск независимого сервиса, то лучше просто использовать для этого контейнер.

Вдобавок эти двое — совершенно разные звери: контейнеры являются подборкой технологий, объединяющих ядро и пространство пользователя,

## Доводы в пользу Casync

Леннарт Пёттеринг [Lennart Pöttering] недавно анонсировал новый проект, *Casync*, разработанный, чтобы облегчить бремя распределения образов файловой системы по сети. В эпоху контейнеров и атомарно обновляемых ОС легко понять, насколько удобен будет подобный инструмент.

*Casync* (content-addressable synchronization — контентно-адресуемая синхронизация) намерен быть эффективнее традиционных инструментов (индиз командной строки при желании могут

добиться того же), особенно при работе с часто обновляемыми образами. Вдохновителями проекта стали такие традиционные инструменты, как *rsync*, *git* и *tar*, и его целью является лучшая, чем у текущих механизмов, работа.

Например, *Docker* использует многоуровневые файловые системы overlay2, хранящие разницу между уровнями. В рабочих системах это хранение бывает ненужным. *OSTree*, *git*-подобная модель, используемая для распространения/обновления

файловых систем, как например, в Endless OS (см. наш обзор в LXF213/214) и flatpak, для сравнения, создают значительный объем излишнего трафика.

Поскольку устройства IoT наконец-то начали серьезно относиться к безопасности, а люди начали вытворять всё более чудные вещи с VM и контейнерами, необходимость в эффективном методе обновления стала вполне очевидна. Читайте больше о *Casync* в блоге Леннарта на <http://0pointer.net/blog/casync-a-tool-for-distributing-file-system-images.html>.

устройств, куда мы можем поместить файловую систему с помощью, например, `mkfs.ext4 /dev/loop0p1`. И смонтировать его следующими командами:

```
$ sudo mkdir /mnt/virtual
```

```
$ sudo mount /dev/loop0p1 /mnt/virtual
```

Воспользуемся *rsync*, чтобы всё скопировать в образ. Некоторые вещи приведут к проблемам, если мы попытаемся скопировать их в VM — например, директории `/dev/`, `/proc/` и `/sys/`; и, возможно, чего-нибудь нам не захочется копировать по чисто прагматичным соображениям, например, большие директории. Измените соответствующим образом опцию `-exclude`:

```
$ sudo rsync -aAXn --info=progress2 /* /mnt/virtual --exclude={/dev/*,/proc/*,/sys/*,/tmp/*,/run/*,/mnt/*,/media*,/home/jonni/.local/share/Steam/*,/home/jonni/.cache/*}
```

### Практические операции

Заметьте, что исключив директорию `/mnt/`, мы избегаем кошмарной рекурсии, когда *rsync* пытается синхронизировать растущую директорию `/mnt/virtual/`. `n` в `-aAXn` велит *rsync* провести тестовый прогон, что всегда неплохо. Если вас устраивает объем данных для переноса, повторите команду без `n`. После завершения можно размонтировать и затем конвертировать файл образа в образ *QEmu* с помощью

```
$ sudo umount /mnt/virtual
```

```
$ qemu-img convert -c -f raw -O qcow p2v.img p2v.qcow
```

Команда *VBoxManage* может выполнить эквивалент *VirtualBox*. Создайте новую виртуальную машину с помощью *virt-manager*, или иного по вашему предпочтению, и присоедините конвертированный

файл образа. Мы сможем его использовать не раньше, чем установим системный загрузчик, поэтому нам нужен live ISO для загрузки. Из среды live введите следующее:

```
$ sudo mount /dev/sda1 /mnt
```

```
$ sudo grub-install /dev/sda --boot-directory=/mnt/boot
```

Затем надо войти через *chroot* в нашу новую файловую систему, чтобы выполнить финальную настройку:

## Возможно, чего-нибудь нам не захочется копировать по прагматичным соображениям.

```
$ sudo mount --bind /dev /mnt/dev
```

```
$ sudo mount --bind /proc /mnt/proc
```

```
$ sudo mount --bind /sys /mnt/sys
```

```
$ sudo chroot /mnt /bin/bash
```

Внутри *chroot* надо создать новый config для *Grub* и обновить `initramfs` для учета (важных) изменений оборудования.

```
$ sudo grub-mkconfig -o /mnt/boot/grub/grub.cfg
```

```
$ sudo update initramfs -u
```

```
$ blkid
```

Последняя команда отобразит UUID наших виртуальных блочных устройств. Вам понадобится обновить `/etc/fstab`, заменив его на соответствующий вашей новой файловой системе `root`.

Теперь надо надеяться, что после перезагрузки всё будет работать. **LXF**

тогда как современная виртуализация в основном является функцией CPU. И опять же, контейнеризация всего подряд тоже имеет свои подводные камни, и люди слишком хотели заняться именно этим.

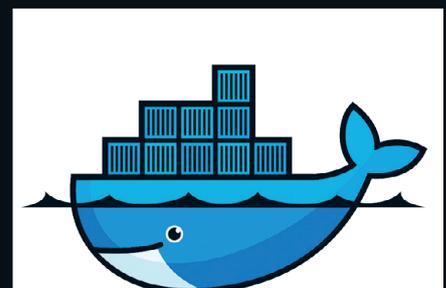
### Узкие места и недостатки

Преимущества безопасности, предлагаемые контейнерами, легко сводятся к нулю, если вдруг сид-минам нужно деизолировать их, чтобы получить доступ ко всем необходимым ресурсам. Очень легко сбиться с пути из-за томов, контейнеров томов и всего прочего, и до того, как вы это даже осознаете, совершенно запутаться в том, чему и что должно быть видно, и спровоцировать появление узких мест, ненужных осложнений или даже уязвимостей.

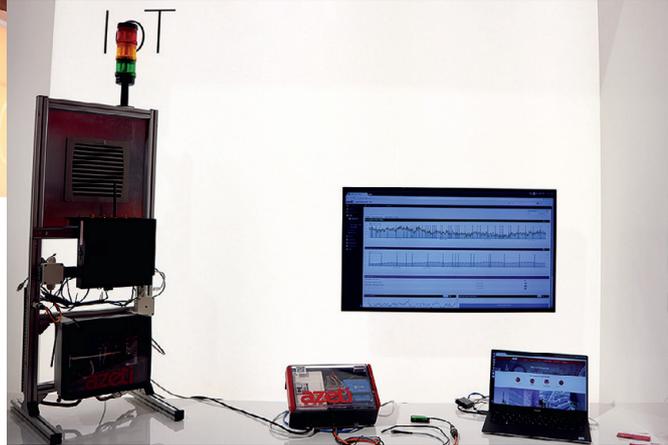
Последняя тенденция в данной области — микросервисы; это естественная эволюция общеизвестной

философии Unix (наличия множества отдельных инструментов, которые отлично выполняют свои небольшие задачи и легко объединяются вместе) насчет использования приложений. Хотя пока что она не является промышленным стандартом, но идея эта вполне понятна и вполне визуализуема в виде, скажем, группы минимальных контейнеров, соединенных между собой и выполняющих ту же работу, которую выполнял традиционный сервер или группа серверов.

Эта модульность упрощает понимание, обновление и тестирование приложений или сервисов. Правильно реализованная, она также улучшает процесс разработки, поскольку команде достаточно понимать только мелкие сервисы, над которыми она работает, и примерную связь между ними и другими сервисами.



► Трудно оценить, насколько Docker обязан своим успехом качеству логотипа компании в виде кита.



# ЖИЗНЬ на краю

Джонни Бидвелл поговорил с представителями Canonical, Майком Беллом и Тибо Руффино, о перспективных разработках в сфере периферийных вычислений.



**Майк Белл [Mike Bell]** — исполнительный вице-президент подразделения Canonical по работе с устройствами и IoT-технологиями. Ранее в 2017 г. (в LXF223) мы обсуждали с ним кое-что из экспонатов Canonical на Mobile World Congress 2017, и у нас не хватило места, чтобы напечатать всё целиком. Теперь хватит и на это, и на новости...

Майк поделится своими ценнейшими знаниями о Snap-пакетах — безопасном и применимом к любому дистрибутиву формату упаковки от Canonical, а также Ubuntu Core — компактном дистрибутиве Linux для устройств IoT. С момента MWC2017 в Canonical произошли изменения — очевидно, кое-кто из наших читателей до сих пор горюет о закрытии Unity, но появились и новые многообещающие партнерские связи в сфере Edge Computing, когда данные обрабатываются

не на каком-то заоблачном центральном сервере, а практически там же, где идет их сбор/генерация.

**Тибо Руффино [Thibaut Rouffineau]**, глава маркетинга, расскажет об этих новшествах и их роли в развитии облачных вычислений, а также сопутствующей концепции Software Defined Everything (SDx).

**Linux Format:** Недавнее известие о приостановке разработки Unity/Mir многие пользователи

настольных систем расценили как утрату идентичности Ubuntu. Но сферы IoT/облака это особо не касается. Не будут ли теперь основные ресурсы направлены туда, и чего нам ждать от нового оптимизированного Canonical?

**Тибо Руффино:** Решение по Unity было непростым, но, вместе с тем, оно обозначило новый фокус Canonical на развитии облачных и IoT-технологий и связанным с этим перераспределением внутренних ресурсов для обеспечения максимального роста. Ubuntu по-прежнему самая широко востребованная облачная платформа как для Public Cloud, так и для OpenStack, и наш облачный бизнес продолжает год от года расти.

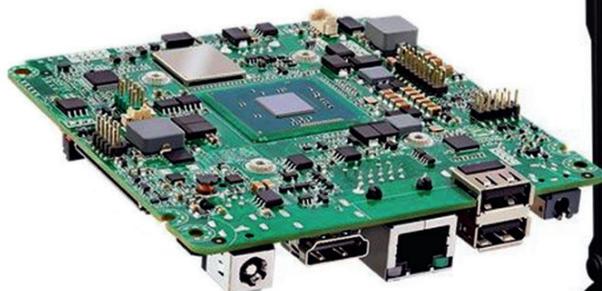
Настольная версия продолжает быть ключевой для бизнеса Canonical и сообщества Ubuntu. Мы много работаем над оптимизацией опыта пользователя, как видно из нашего недавнего опроса, где мы попросили высказать свои пожелания. Мы с радостью принимаем вызов создать новый особый облик Ubuntu в предстоящем обновлении LTS в 2018 г. И, что еще важнее, всё сообщество предвкушает это новое и очень активно включается в происходящие изменения.

В сфере IoT у нас меньше опыта, но на сегодняшний день имеется много положительных результатов. Здесь многое строится вокруг Ubuntu Core, минимальной ОС, разработанной специально для встроенных устройств. Мы наблюдаем активное внедрение IoT Ubuntu во многих отраслях, включая робототехнику, цифровые вывески, пограничные шлюзы и самоуправляемые автомобили. В перспективе, мы продолжим продвигать Ubuntu Core и snap-пакеты в сфере встроенных систем при поддержке ключевых стратегических партнеров.

В частности, мы будем фокусироваться на промышленном рынке IoT, поскольку мы видим, что организации переходят с OT на IT, связывая всё большее количество промышленных активов. Мы уже работали над рядом проектов, нередко совместно с Dell, по внедрению Ubuntu Core на их компьютерах с пограничными шлюзами, для увеличения обработки данных и анализа на границе автономных систем.

**LXF:** Canonical недавно присоединились к проекту Linux Foundation EdgeX Foundry. Расскажите немного о фрагментации в сфере IoT и о том, как этот проект поможет с ней справиться.

**ТР:** IoT — это, по существу, новое имя для того, что прежде называлось встроенным или M2M-



» Ubuntu Core работает на устройствах всех форм и размеров (в основном небольших).



» Майк Белл: «Ubuntu Core проектировалась с учетом безопасности».

пространством. Встроенная разработка существует уже много десятилетий, накопив огромное наследие кода, протоколов, узкоспециальных решений, сложных и долгих процессов создания или распространения ПО. Короче говоря, это разобщенная экосистема с ощутимым отсутствием стандартизации или реально действующих библиотек стандар-

приводя к росту числа мелких проектов и усложнению структуры пространства IoT. Что негативно скажется, например, на сроках разработки и безопасности, то есть двух ключевых факторах, способных отвлечь крупные компании от внедрения IoT, либо поставить под угрозу безопасность их IoT-приложений.

## РАЗРАБОТКА — И БИЗНЕС В УМЕ

**Создавая операционную систему, нужно быть уверенным, что она соответствует своей цели.**

тов. Поскольку, на волне всеобщей популярности, всё больше людей спешат перейти на IoT, им важно быстрее вывести продукт на рынок, чтобы ПО быстрее попало в руки пользователей...

К сожалению, нам очевидно, что традиционная раздробленность сферы встроенных технологий может стать препятствием для многих компаний,

Как один из основателей компании, могу сказать, что мы в Canonical приветствуем внедрение EdgeX Foundry, видя в нем решение части этих проблем. Создание единой открытой структуры и экосистемы компаний, предлагающих совместимые компоненты plug-and-play, позволит компаниям сконцентрироваться на повышении доходности IoT, не тратя время на интеграцию и развертывание своих решений. Эти преимущества распространяются на всех участников экосистемы IoT, поставщиков оборудования и программного обеспечения для системных интеграторов и, самое главное, конечного потребителя.

**LXF:** Давайте поговорим о безопасности, особенно безопасности IoT. Судя по таким вещам, как вредоносное ПО Mirai и массивные DDoS-атаки, напрасно надеяться, что пользователи массово начнут менять пароли по умолчанию, и, как давно принято считать, по доброй воле ни за что не обновят ПО. Как Ubuntu Core и snap-пакеты решают эту проблему?

**Майк Белл:** Продумывая дизайн Ubuntu Core, мы думали о безопасности изначально. Говоря о существующих проблемах, одна из них — то, что устройство может довольно долго находиться на рынке. Хотя IoT привыкли связывать с бытовой электроникой, и об этом много шумят, но приложения, на которых строятся важные промышленные



проекты, зачастую рассчитаны на устройства, срок службы которых составляет до 10 лет.

Принимая это в расчет, вы должны быть уверены, что создаваемая вами операционная система подходит для этой цели. Поэтому мы, прежде всего, исходили из соображений безопасности, и я имею в виду не просто безопасность на уровне эксплуатации. Мы, на самом деле, разработали весь продукт таким образом, чтобы у нас был уровень изоляции и замкнутости, и любую часть стека, от ядра до приложений, можно было обновить по беспроводной связи.

В snap-пакетах используется особая технология ядра под названием AppArmor; она опирается на механизмы дискреционного доступа в Linux и становится обязательным форматом. Это означает, что ее поддержка должна быть прописана во всем, к чему вам требуется получить доступ — будь то файлы или сокет. В противном случае происходит отмена операции. Таким образом нам удается минимизировать угрозы, которые впоследствии могут проявить себя в ПО. Мы берем само ядро и делаем из него snap-пакет.

Canonical поддерживает свои LTS-версии пять лет, но некоторым нашим промышленным клиентам мы теперь предоставляем 10 лет, в плане выхода обновлений. По сути, мы будем предоставлять обновляемый snap ядра на пять лет из коробки с возможностью продления для некоторых клиентов. Как правило, мы выпускаем до 30 обновлений ядра и операционной системы для устранения критических уязвимостей.

**LXF:** То есть модель подписки или, по крайней мере, долгосрочное соглашение об обслуживании, такое как у вас с промышленными партнерами, лучше подходит для устройств IoT, раз есть такие напасти, которые надо устранять постоянно?

**МБ:** Думаю, что ответ здесь будет двоякий. Если мы говорим о промышленных приложениях, это

действительно так. Понятно, что все хотят платить как можно меньше, но предполагается, что в них обновления безопасности потребуются в течение всего периода поддержки продукта.

Что касается потребительского рынка, здесь реальность такова, что люди хотят как можно быстрее продвинуто продукты и сервисы на рынке. И здесь гораздо меньше задумываются о том, что это будут за устройства и как долго они проживут на рынке, или в чьем-то доме, или на работе. Так что на рынке бытовой электроники мало кто рассматривает вариант пяти- или десятилетней поддержки. Мы надеемся, что они выберут нечто работающее на Ubuntu Core из коробки, понимая, что с этой системой они будут получать обновления пять лет, и это куда больше, чем во многих других ОС.

**LXF:** Наличие таких продвинутых угроз, как ShellShock и Heartbleed, формирует представление, что проекты Open Source не слишком заботятся о безопасности. Что, в целом, неверно, о чем свидетельствуют принципы разработки и обновления Ubuntu Core. Но всегда есть к чему стремиться, особенно если учесть, сколько денег приносят продукты с открытым исходным кодом Силиконовой долине. Чем, в этой связи, корпорации и сообщество в целом могут помочь сделать лучше?

**МБ:** На самом деле, сообщество, куда входят и представители отрасли, и люди из научной сферы, и сами пользователи, вносит огромный вклад в развитие безопасности. Мы создаем огромную базу кода, которая разрабатывается и совершенствуется годами. Но всегда найдутся хитрецы, которые найдут в ней лазейки, так что речь идет о том, как быстро они устраняются и как быстро вы донесете это обновление до своих клиентов.

Я думаю, что нет такой ОС, которая всегда будет на 100% безопасной. Можно утверждать, что она является таковой в данный момент времени, и мы стараемся делать всё возможное, чтобы при обнаружении CVE заплатка была выпущена и опубликована как можно скорее. Мы взяли это за правило, и в случае с недавней уязвимостью Dirty Cow мы были одной из первых ОС, выпустивших исправления для наших клиентов. Объем исходного кода сейчас огромен, и на поиск уязвимостей уходит куча времени.

Мне кажется, что сейчас всё идет правильно. По-моему, гораздо разумнее разрабатывать ПО с учетом безопасности, нежели развивать ее потом. В Ubuntu Core подход как раз такой: использование AppArmor дает защиту от некоторых уязвимостей.

От всего защититься невозможно, но если вы используете приложение на устройстве с Ubuntu Core, вы будете иметь доступ только к ресурсам этого приложения и не сможете использовать это устройство для того, для чего оно не предназначено.

Взять хотя бы все эти Wi-Fi-камеры, переносимые и включенные в бот-сети. Это действительно хороший пример, как независимые поставщики могли бы предотвратить такое использование приложения, установив ограничение на ресурсы. Это помогает уменьшить — может быть, не всегда предотвратить, но уменьшить — некоторые риски, связанные с кибер-угрозами.

**LXF:** Snap-пакеты и инструменты управления ими теперь доступны для всех традиционных настольных и серверных версий Ubuntu. Значит ли это, что они заменят .Deb и традиционные методы управления пакетами?

**МБ:** С тех пор, как мы добавили snap-пакеты в настольный Ubuntu и на сервер, мы видим, как сообщества, как с открытым, так и с закрытым кодом, стали распространять свое ПО в этом формате. У него масса преимуществ при упаковке настольных и облачных приложений, в том числе возможность изолирования зависимостей. Когда у вас есть уже готовый дистрибутив, интегрировать туда новейшее ПО бывает довольно сложно. Собирая зависимости и изолируя их, мы можем запускать приложения, которые не были предназначены для работы в этом дистрибутиве на момент его выхода.

**LXF:** Иногда всё обходится добавлением PPA, но гораздо чаще людям приходится что-то компилировать, пытаться встроить это в среду других приложений, также обновленных вручную — и в результате выходит полная каша. И здорово, что snap-пакеты (и flatpak) помогают сделать это гораздо опрятнее. Но какого объема будут такие snap'ы, если они вмещают всё древо зависимостей?

**МБ:** Да, но когда вы устанавливаете что-либо впервые, вам всё равно всё это понадобится на вашем ПК. Мы стали применять на наших серверах дельта-кодирование, так что когда в snap вносятся какие-то изменения, мы будем обновлять только эту часть кода. Как правило, дистрибутив обновляется в пределах 10%. Это зависит от технологии: некоторые лучше подходят для дельта-кодирования. Но практически все основные технологии разработки, которые используются в Linux, как правило, попадают под эти 10%. Например, если у кого-то есть JRE, и они обновляют только свой пакет, а не всю JRE, то мы поменяем только дельта-элементы, а не весь дистрибутив. Это выгоднее и в плане времени, и для самой архитектуры сервера. И упрощает ситуацию, когда у вас может возникнуть древо зависимостей из других двоичных программ. Использование дельта-алгоритмов позволяет вносить изменения только в само приложение, минуя зависимости.

**LXF:** Очевидно, что между технологиями Snap и Flatpak есть некоторые пересечения. Вы рассматриваете их как конкурирующие стандарты? Или Flatpak лучше подходит для настольных приложений, тогда как Snap — более универсальная вещь? Станут ли они в будущем (частично) совместимыми?

► Мы так и не узнали, что это (см. LXF223).



**TP:** Snap-пакеты изначально разрабатывались как многофункциональное решение, применимое от периферийных устройств до облака, тогда как Flatpak больше ориентирован на рабочий стол. Чем и объясняются некоторые технические различия между ними. Snap стали лучшим вариантом для разработчиков, использующих чрезвычайно популярную среду *Electron* для настольных приложений Linux. Также их использует проект Kubernetes, одно из самых популярных имен в облаке. Появляется очень много интересных интернет-продуктов, построенных на Ubuntu Core и snap.

В конечном счете, решение о выборе формата будет за поставщиками ПО. И очень обнадеживает, что если раньше это были только Ubuntu, то теперь всё больше производителей ПО и дистрибутивов Linux переходят на snap-пакеты.

**LXF:** Amazon запустила Greengrass — приложение, работающее на подключенных устройствах, что позволяет производить больше вычислений на периферии и улучшить локальную связь между устройствами IoT. Greengrass может работать в разных окружениях Linux. Это тоже стало возможным благодаря Canonical, которые сделали его доступным в виде snap?

**TP:** Да, благодаря тому что Greengrass доступен в виде snap-пакета, его можно запускать на других дистрибутивах Linux. Однако работа с ним непосредственно в Ubuntu Core имеет ряд особых пре-

## НАДЕЖНОСТЬ ЯДРА

### С Ubuntu Core ваши клиенты в полной безопасности...

имущества. Например, Ubuntu Core сертифицирован для использования с Greengrass и является предустановленной ОС в ряде периферийных устройств.

Кроме того, в Ubuntu Core теперь есть единый и многократно проверенный механизм обновления «по воздуху». Все, кто использует периферийное устройство на Ubuntu Core, могут быть спокойны, зная, что они будут автоматически защищены последними обновлениями безопасности по мере их появления. Совсем другое дело, чем использовать какой-нибудь более локальный, менее поддерживаемый или проприетарный дистрибутив Linux. Ubuntu Core обеспечивает гораздо лучшую защиту для своих конечных пользователей и их подключенных устройств.

**LXF:** Greengrass позволяет больше обработки, прежде выполнявшейся в облаке, производить локально. В чем плюсы переноса этих некогда облачных операций на землю?

**TP:** Важным моментом в отношении нового продукта Amazon Greengrass является то, что это отнюдь не альтернатива для всех облачных операций. Мы рассматриваем AWS Greengrass (и периферийные технологии в целом) как прекрасное дополнение к облаку, позволяющее подключать локальные



Тибо Руффино называет себя евангелистом Ubuntu Core и IoT.

сервисы там, где это более уместно, и направлять самые нужные данные в облако.

Даже с учетом того, что значительная доля обработки в некоторых приложениях уйдет на периферийные устройства, мы предполагаем, что по мере роста востребованности IoT облако и периферия будут развиваться в тандеме. В некоторых случаях использование облака будет уместно только посредством периферии. В целом, это взаимовыгодная ситуация.

При этом, у периферийных вычислений тоже есть масса преимуществ. Есть фактор скорости: периферийные устройства часто гораздо быстрее реагируют на критические события и выполняют надлежащие действия, что актуально в особо важных системах (например, заводских). Они также могут агрегировать и переводить данные, отправляя в облако только самое нужное, что позволит сэкономить на пропускной способности и затратах на облачные вычисления. Они также могут защитить от простоя подключения. Но эти факторы должны учитываться наряду с более широкими возможностями облака.

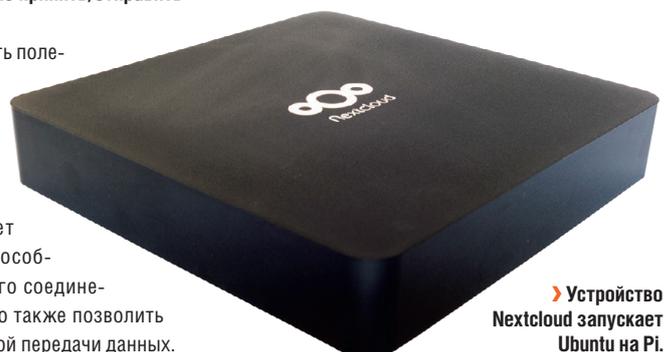
**LXF:** То есть, целый ряд задач (обработка гигантских наборов данных, связь с удаленными серверами) всё равно придется выполнять в облаке, где Greengrass напрямую не помогает. Но я предполагаю, что он себя оправдывает, скажем, в ситуации, когда возможности подключения ограничены и невозможно принять/отправить сообщения как есть?

**TP:** Да, Greengrass может быть полезен, когда нужно компенсировать ограниченные возможности подключения. При том, что вся обработка производится локально, устройство может ограничить пропускную способность, скажем, спутникового соединения на нефтяной вышке. Это также позволит решить проблему с задержкой передачи данных.

Что касается гигантских наборов данных, опять же, есть определенные ситуации, когда более сложный анализ логичнее провести в облаке, возможно, со ссылкой на другие наборы данных или информацию, генерируемую в реальном времени вне сети на периферийных устройствах. Однако с помощью тех же устройств можно разбить эти большие наборы данных, прежде чем они попадут в облако, устранив дублирование и лишние элементы, таким образом, более рационально расходуя ценное время облачного сервера.

**LXF:** В нашем разговоре по телефону Майк употребил фразу «ПО определяет всё [Software Defined Everything]». Как Greengrass вписывается в эту идею?

**TP:** Greengrass является одной из реализаций общей концепции Canonical “Software Defined Everything”. Я подозреваю, что дни проприетарных периферийных устройств, в целом, сочтены. Показательно, что лучшее из того, что мы сегодня видим, основано на аппаратных средствах общего назначения. Используя Ubuntu Core, специальную ОС от Canonical для IoT, можно улучшить функциональность данного оборудования с помощью ПО: через snap-пакеты. Можно даже получать новые приложения через магазин приложений. Это идеальное воплощение программного обеспечения IoT. То, на что раньше тратилось несколько месяцев (определение аппаратного и встроенного программного обеспечения), теперь делается за несколько дней. **LXF**



Устройство Nextcloud запускает Ubuntu на Pi.



# LINUX vs BSD: БОЙ ЗА ТИТУЛ

Стараясь быть непредвзятым, Маянк Шарма сталкивает двух гигантов открытого кода в попытке закрыть вопрос о превосходстве (по крайней мере, так он надеется).

**Р**едко ли вам приходилось слышать, как валят в одну кучу Linux и BSD? Мол, какая разница, они оба с открытым кодом; а более информированные указывают на их общее происхождение. Да и мы тоже неоднократно их уравнивали, особенно при поверхностном сравнении. Но если заглянуть внутрь, то сходство очень скоро уступит дорогу массе различий.

Хотя ОС типа BSD и не привлекали особого внимания популярных изданий, но их любят за стабильность, надежность и безопасность, и они весьма популярны среди системных администраторов. Иными словами, на них

легко поставить популярные приложения с открытым кодом и использовать BSD для выполнения повседневных настольных задач — и даже для того, чтобы играть. Кроме того, стоит отметить их сильное и преданное сообщество разработчиков и пользователей.

Современные BSD восходят к 1970-м. BSD означает Berkeley Software Distribution. Так назывался инструментарий для улучшений в UNIX, созданный

в калифорнийском Университете Беркли. В противоположность UNIX, разработанному компанией Bell Labs, BSD разрабатывался студентами и преподавателями университета и распространялся в виде пакета программных улучшений для UNIX, которые делали его полезным в реальном мире, за стенами исследовательской лаборатории.

Со временем BSD эволюционировал и заменил все части UNIX, став удобной в работе самостоятельной операционной системой. Современный парк дистрибутивов BSD — это целое семейство операционных систем, ведущих свою родословную от BSD.

**Если заглянуть внутрь Linux и BSD, то сходство очень скоро уступит дорогу массе различий.**

## Средства достижения цели

Прежде чем рассматривать разные аспекты BSD и Linux, способствующие их различиям, давайте рассмотрим их происхождение и то, как оно повлияло на их сходство.

Linux — это ядро, разработанное Линусом Торвалдсом с целью замены ориентированной на ИТ-образование, но очень ограниченной системы Minix в Университете Хельсинки, Финляндия. Это ядро в сочетании с некоторыми другими компонентами — в первую очередь, пакетом утилит GNU — превратилось в UNIX-подобную операционную систему. Поскольку Linux был написан с нуля и не является потомком Unix, он значительно отличается от других систем, имеющих более тесные связи с UNIX.

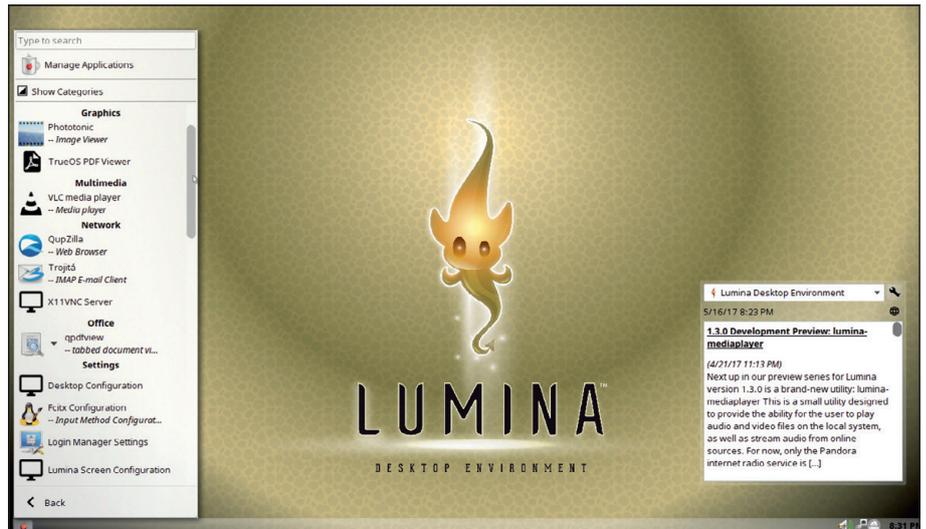
BSD, напротив, поддерживает теснейшие связи с прошлым, в лице одной из самых выдающихся версий, FreeBSD, которая разработана как клон UNIX с открытой лицензией. Разработчики BSD следуют образу действий UNIX, что привело к некоторым техническим различиям между этими двумя операционными системами. Более того, подход UNIX продолжает влиять на все будущие разработки в стране BSD.

Сходство между BSD и Linux объясняется тем, что обе эти ОС по природе своей похожи на UNIX. Это влияет на всё, от лежащей в основе архитектуры системы и ее цели до структуры ОС и взаимодействия ее компонентов. И Linux, и BSD — POSIX-совместимые операционные системы, что помогает стандартизировать их структуру и архитектуру. Например, у них обоих основным средством взаимодействия с лежащими в основе компонентами является интерфейс командной строки, и иерархии файловой системы похожи.

## Гармоничное существование

Экосистема Linux до отказа набита дистрибутивами, большинство которых являются производными одного или другого дистрибутива-тяжеловеса, например, Debian, Fedora, Ubuntu, Arch или даже Slackware. Помимо них, растет число дистрибутивов, написанных с нуля.

Исходный BSD, возможно, уже канул в Лету, но термин по-прежнему применяется к букету современных его производных. Три самых известных потомка BSD, используемых сегодня — это FreeBSD, OpenBSD и NetBSD. Большинство ОС BSD



➤ Легковесный рабочий стол Lumina создан TrueOS для систем на базе BSD.

имеют открытый код и доступны для скачивания бесплатно по Лицензии BSD; заметное исключение составляет Mac OS X со своими проприетарными дополнениями.

Изобилие дистрибутивов Linux пользователи BSD часто рассматривают как отражение дезорганизации и раскола сообщества. Однако этот аргумент не выдерживает критики, поскольку все ди-

## Аппаратные хиты

Говоря в общем, операционные системы с открытым кодом не получают такого же уровня официальной поддержки со стороны производителей оборудования, как проприетарные системы вроде Windows и Mac OS X. Хотя сейчас производители оборудования более благосклонно смотрят на ОС с открытым кодом, всё равно будет куда про-

## Аппаратная поддержка BSD — на том уровне, где была аппаратная поддержка Linux на рубеже веков.

стрибутивы используют то же самое ядро Linux и пользовательские утилиты. Более того, дистрибутивы двоично-совместимы и часто имеют одинаковые основные ресурсы, например, код и структуру — и даже разработчиков.

А вот сотрудничество среди вариантов BSD является весьма щекотливым вопросом. Конечно, можно отыскать примеры, когда разные проекты BSD объединяли свои ресурсы, но это по-прежнему остается скорее исключением, чем правилом. Фактически, многие популярные BSD появились в итоге очень громогласных споров и разногласий среди разработчиков.

ще использовать новое оборудование с популярной проприетарной ОС, чем с популярной ОС с открытым кодом.

Среди двух основных ОС с открытым кодом именно Linux опережает BSD в плане привлечения внимания со стороны производителей оборудования. Некоторые производители, например, Nvidia, выпускают драйверы для таких разновидностей BSD, как FreeBSD, но большая часть аппаратной поддержки для BSD идет от сообщества разработчиков, которые без усталости портируют и подвергают обратному инжинирингу драйверы для популярных устройств. »

## Лицензия на убийство

Еще одно значимое различие между Linux и BSD — лицензирование. Ядро Linux, приложения GNU и многие программы из мира Linux выходят под лицензией GNU General Public License, она же — GPL. Ее часто именуют копиелефтом, в противоположность копирайту, потому что GPL обеспечивает свободу просмотра, распространения и изменения исходного кода, требуя подчинения всех производных тем же основным положениям лицензии.

В основном GPL требует, чтобы разработчики Linux выпускали исходный код своих модификаций.

BSD же используют лицензию BSD License, которая, говоря попросту, разрешает не делать модификации открытыми, если так пожелает разработчик. Лицензия BSD предоставляет разработчикам больше свободы, чем GPL, поскольку не требует, чтобы производная работа выполняла лицензионные требования кода оригинала. Поэтому любой может использовать, распространять и даже модифицировать код, не будучи обязанным делать изменения открытыми или выпускать исходник их производного кода.

Если взглянуть на лицензии GPL и BSD под другим углом, можно сказать, что первая создана для защиты прав пользователей, а вторая скорее нацелена на интересы разработчиков. Интересно, что проект OpenBSD приложил значительные усилия на удаление из релиза OpenBSD кода под GPL. По историческим причинам базовая система OpenBSD по-прежнему включает несколько компонентов под GPL, но твердо намерена не принимать в базовую систему дополнительных программ, связанных положениями лицензии GPL.

Чтобы представить это в перспективе, обратите внимание на сравнение беспроводных драйверов с открытым кодом — [https://en.wikipedia.org/wiki/Comparison\\_of\\_open-source\\_wireless\\_drivers](https://en.wikipedia.org/wiki/Comparison_of_open-source_wireless_drivers)—и вы обнаружите: при том, что именитые производители, такие как Qualcomm, Broadcom, Intel, Marvell, Realtek, Nokia и Texas Instruments, всё-таки пишут драйверы устройств для Linux, BSD остается позабытой о себе самостоятельно.

Справедливо будет сделать вывод, что аппаратная поддержка для BSD находится на том же уровне, на котором для Linux она была на рубеже столетий [Ред.: — Когда всё это было “404 not found”]. Разницу между Linux и BSD оценить трудно, но если вы покупаете настольное оборудование, то шансы, что оно заработает сразу, у Linux значительно выше, чем у BSD.

## Программные проблемы

С точки зрения пользователя, одним из ключевых различий между FreeBSD и большинством дистрибутивов Linux является форма и доступность программ. Хотя в Linux ничто не препятствует компиляции программ из исходника, типовая установка Linux обычно содержит только прекомпилированные двоичные пакеты.

BSD, напротив, обычно содержит одновременно и прекомпилированные пакеты, и пакеты, устанавливаемые из исходника. Причина в том, что BSD гордится своей системой сборки для компиляции и установки программ из исходника через



➤ Некоторые BSD, например, FreeBSD и NetBSD, двоично-совместимы с Linux, и их пользователи могут устанавливать и запускать почти любой бинарник Linux.

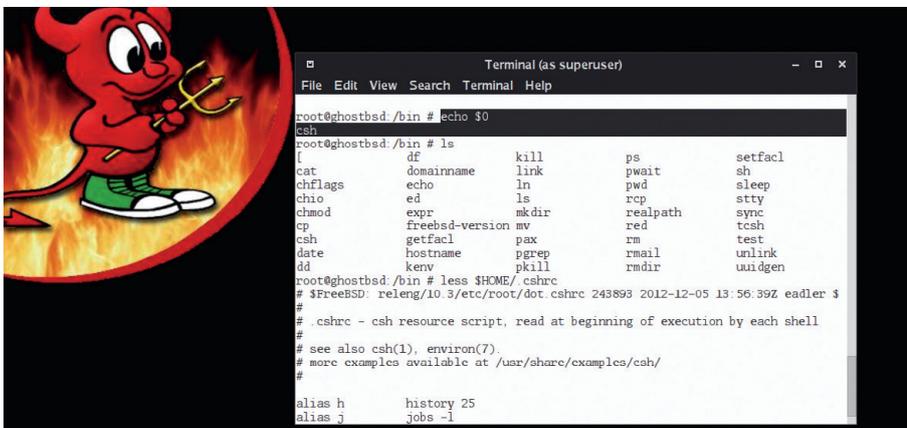
систему, известную как совокупность портов [ports collection].

Считайте совокупность портов набором сборочных файлов и заплат-патчей, которые BSD используют для компиляции пакетов. Иными словами, основное назначение системы портов — автоматизировать процесс компиляции и установки пакетов. Совокупность портов FreeBSD была адаптирована NetBSD, OpenBSD и DragonFly BSD для их собственных систем управления пакетами на основе портов. Ныне совокупность портов FreeBSD является самой

обширной из всех и содержит более 27 000 портов. Говоря в целом, в мире BSD пакетные версии программ создаются из системы портов. Это делает BSD операционной системой с упором на исходник, где пакеты доступны чисто удобства ради.

Хотя горячие поклонники BSD свято верят в систему совокупности портов, среднего линуксоида она может не особенно впечатлить. Те, кто компилировал программы в Linux, определенно оценят удобство стандартизированного процесса компиляции, предлагаемого портами. Но если вы устанавливали только двоичные пакеты, то пользователям BSD придется изрядно потрудиться, убеждая вас в достоинствах системы портов.

В отличие от портов, большинство дистрибутивов Linux предлагают двоичные программы в основном через онлайн-репозитории. Большинство дистрибутивов имеет собственный набор репозиторий, которые можно просмотреть с помощью простых в использовании менеджеров пакетов, которые скачивают и устанавливают пакеты. У основных дистрибутивов имеются хорошо отлаженные процедуры создания пакетов, и все они отводят значительные ресурсы на поддержку своих репозиторий, чтобы устранять ошибки и делать доступными обновления. Модель репозиторий обеспечивает готовность к работе, надежность, безопасность и своевременное обновление установок Linux.



➤ По умолчанию BSD используют оболочку *csh*, и их директория */bin* содержит только исполняемые файлы для компонентов в базовой системе.

## Мушкетеры BSD

По данным обзора использования BSD, среди BSD чаще всего выбирают FreeBSD. Первый релиз этой операционной системы вышел в 1993 г., и ее популярность привела к появлению нескольких производных. Одним из таких потомков является TrueOS (бывший PC-BSD), который сделал себе имя, предложив знаменитую стабильность FreeBSD обычным пользователям настольных систем.

Одной из самых безопасных ОС считается OpenBSD. Благодаря своему строжайшему аудиту кода и модели разработки, которая во главу угла ставит

безопасность, она является популярной опцией для таких областей обеспечения безопасности, как брандмауэры, системы обнаружения вторжений и серверы общего назначения. Она разработана для специализированного использования, поэтому аппаратная поддержка OpenBSD в большей мере ориентирована на оборудования уровня предприятия и даже виртуальное оборудование.

А вот дистрибутив NetBSD (существующий с начала 1990-х) в соответствии со своим слоганом «Конечно, NetBSD здесь работает» отлично переносим

и работает на более чем 50 аппаратных платформах на 15 архитектурах процессоров. Еще одна важная функция этой ОС в том, что в отличие от типичных релизов настольного Linux, выпускаемых примерно раз в полгода, новые релизы NetBSD выходят довольно нечасто. Это вполне подходит ее основным пользователям, которые не хотят часто дергать свои серверы для обновления, и отнюдь не означает, что у вас будут устаревшие программы: система управления пакетами, *pkgsrc*, отслеживает самые свежие версии вышележащего [upstream] ПО.

## Внутренняя сила

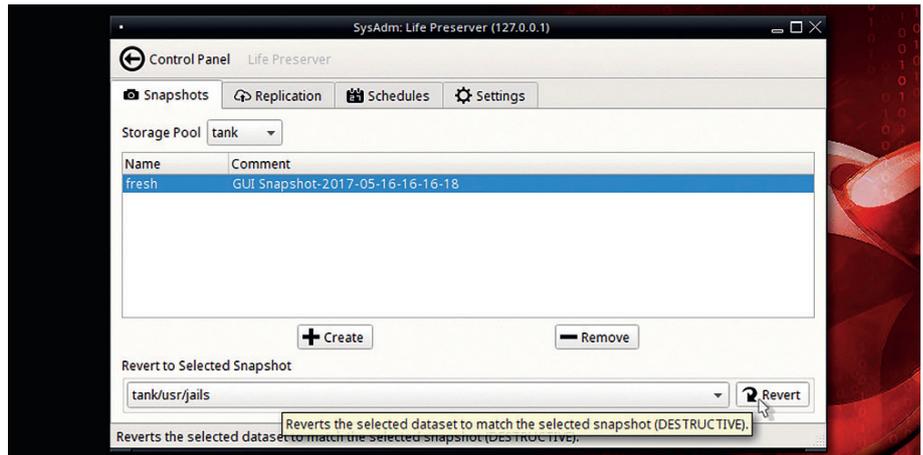
Одно из основных отличий BSD и Linux — их конструкция. В чисто техническом смысле Linux — это чисто ядро. Для создания пригодной к работе операционной системы каждый дистрибутив Linux присоединяет к ядру набор библиотек, инструментов и программ. BSD, напротив, разрабатывает ядро и базовую операционную систему вместе, как единое целое.

Хотя для непосвященных это может показаться не слишком большим отличием, это помогает определить, как вы взаимодействуете и управляете каждой системой, и является одной из причин знаменитой стабильности BSD по сравнению с Linux. Причина в том, что в мире Linux дистрибутив собирает основную группу пакетов и обеспечивает их соответствие и взаимодействие друг с другом. Однако их интеграция зависит от умений и опыта разработчиков дистрибутива, которые соединяют их вместе. Поскольку это не очень отличается от установки пакетов, многие дистрибутивы также обновляют основные пакеты через репозитории.

И наоборот, такие BSD ОС, как FreeBSD, поддерживают целую основную операционную систему. Сюда входит ядро и набор основных программ, которые разработаны совместно и поддерживаются в совокупности. В отличие от Linux, в BSD отдельные компоненты, являющиеся частью этого основного набора, не самая тривиальная задача. Основа системы BSD обновляется как единое целое независимо, а опциональные программы можно обновлять отдельно. Такая структура позволяет разработчикам BSD обеспечить тесную интеграцию, в результате которой получается более стабильный продукт.

В Linux же основные пакеты распространяются через специальный репозиторий, чтобы позволить разработчикам сосредоточиться на конкретном подмножестве программ, доступных в репозиториях. Хотя эти основные пакеты очень строго тестируются, само по себе это не обеспечит стабильности системы, как в ОС BSD.

Некоторые считают, что обновление в Linux основных компонентов дистрибутива и отдельных пакетов — лучший и более эффективный способ обновления, чем система BSD, которая требует, чтобы пользователь обновил базовую систему отдельно от остальной части установки. Однако мы видели слишком много рабочих компьютеров



➤ Одна из славнейших функций файловой системы BSD по умолчанию, ZFS — умение делать моментальные снимки, и тут есть ряд преимуществ перед полной резервной копией.

Linux, спалившихся в итоге обновления, чтобы назвать более простой процесс обновления в Linux подлинным удобством.

В продолжение уже упомянутого отличия сборки этих ОС, некоторые аспекты базовой системы BSD, такие как OpenSSH и даже некоторые утилиты GNU, разрабатываются на стороне. Однако они тоже ассимилированы в базовую систему. Это объясняет, почему некоторые утилиты, являющиеся общими для Linux и системы BSD, функционируют в этих двух системах немного по-разному.

## Заключение

У BSD и Linux много общего — и вдвое больше различий. У всех популярных BSD есть собственные сильные стороны. Например, OpenBSD известна своей безопасностью, а NetBSD — способностью ладить с большим числом архитектур.

Те, кто привык к BSD, также мигмом перечислят ряд операционных преимуществ перед Linux. Часто считается, что в BSD лучшая иерархия файловой системы, и пользователи утверждают, что, в отличие от Linux, вы не заблудитесь в поисках файла настройки. Еще одно преимущество, на которое часто ссылаются, в том, что BSD не устанавливает ненужных сервисов, которые работают в фоне.

Однако есть области, где BSD не дотягивают до Linux. Официальная поддержка производителей оборудования — один из аспектов, где Linux значительно опережает BSD. Кроме того, настольный

Linux более заметен, чем настольный BSD. Несмотря на наличие специально настроенных настольных вариантов, дружелюбных к пользователю, в BSD не сумели разрушить стереотип, что это — ОС для умников. BSD по-прежнему известны своей надежностью, и они нашли себя на серверах и встраиваемых системах.

Однако мы можем помочь вам выбрать нужную ОС для вашего проекта. Это потому, что после четырех страниц технического сравнения выбор сводится к одному совершенно не техническому критерию — удобству. Если вам комфортнее в Linux, мы бы не советовали создавать себе проблемы и переходить на BSD. Точно так же, мы не надеемся впечатлить преданных пользователей BSD и убедить их перейти на Linux. От небольших встраиваемых устройств до крупномасштабных серверов и всего, что находится между ними, обе эти ОС являются вполне зрелыми и достаточно гибкими, чтобы подойти для любой области использования.

Поэтому да, в общем и целом мы не настолько наивны, чтобы игнорировать достоинства BSD, и не настолько отважны, чтобы рекомендовать их вместо Linux для всех областей применения. Судя по всему, борьба продолжается. LXF



## Популярные продукты на базе BSD

На базе BSD есть множество продуктов, и с открытым кодом, и коммерческих, благодаря ее техническим достоинствам и удобной лицензии. Среди популярных продуктов с открытым кодом — такие дистрибутивы NAS, как FreeNAS и NAS4Free, и проекты брандмауэров, например, встраиваемый дистрибутив m0n0wall и его ответвление для обычных компьютеров, pfSense.

Далее идет проект с открытым кодом Darwin от Apple; он создает основные компоненты проприетарных ОС компании, OS X и iOS. Darwin использует

код, полученный из BSD, который также взят за основу файловых систем и сетевых устройств OS X.

Существуют и другие коммерческие продукты на базе BSD, от мультинациональных производителей оборудования и ПО, таких, как массив iSCSI SAN от Dell. Silicon Graphics International также использует FreeBSD в своих массивах дисков ArcFiniti MAID (сокращение от Massive Array of Idle Drives), и то же самое делает Sony в своих бестселлерах PlayStation 3 и PlayStation 4, а также в игровом устройстве PlayStation Vita.

Самой популярной среди BSD является FreeBSD, и на ней работают как продукты с открытым кодом, так и проприетарные продукты. На FreeBSD основан JUNOS от Juniper Network; FreeBSD также применяется в комплексе Netflix Open Connect. На FreeBSD работает популярное приложение для обмена сообщениями WhatsApp; и Ян Кум (выходец из бывшего СССР), сооснователь проекта WhatsApp и давний пользователь FreeBSD, заработав миллиарды долларов, пожертвовал один миллион этому проекту с открытым кодом.

# Покупаем Linux-ноутбук



## Вам нужен ноутбук с установленным Linux? Нейл Мор желает вам удачи и предупреждает о проблемах мира ноутбуков.

**И**так, вы хотите купить ноутбук с Linux? Вот вам статья, где мы пойдём и будем изводить недоумевающих бедняг-продавцов в точках розницы вопросами о Linux. Вот посмеёмся-то... Нет, мы этого не сделаем, потому что так нечестно, и в итоге вы всегда найдёте осведомленного о Linux,

и он просто опечалится, потому что с политикой сети продаж ничего не поделаешь. *(Вздых.)*

Вместо этого мы рассмотрим, какие есть реальные опции и какие подходы вы можете выбрать, установив систему сами, чтобы создать действительно открытое оборудование. Конечно, есть решения вроде Chromebook и устройства Android, но многие пользователи Linux воспринимают их неприятно, поскольку они, может, и применяют Linux, но являются не настоящими дистрибутивами GNU/Linux, а чем-то вроде незаконнорожденных версий от Google, без подлинной свободы традиционных дистрибутивов с элементами GNU. Это еще одно подтверждение жизнеспособности GPL в сохранении систем свободными от внедрения проприетарных ловушек, хотя они и с открытым кодом.

Мы отнюдь не намерены игнорировать тех, кто стал настоящими гигантами на потребительском рынке. Как вам известно, Android — самая популярная ОС в мире (возможно, из всех когда-либо

существовавших), и хромбуки начинают составлять настоящую конкуренцию на рынке ноутбуков, отрезая такой же большой кусок, как у Apple. Однако наш основной интерес лежит в области обычных ноутбуков. Мы не собираемся рассматривать игровые модели, поскольку их встроенные графические карты способны вызывать проблемы. Поэтому мы бы рекомендовали избегать всего, что содержит GPU от AMD или Nvidia. Есть специальные сборки драйверов, такие как Bumblebee для Nvidia GPU, однако это похоже на приобретение поросы из-за отсутствия забот. Если вы хотите играть, используйте настольный ПК.

Мы также взглянем на некоторые интересные краудфандинговые модели, которые переопределяют способы изготовления ноутбуков. Однако, конечно, злорадный секрет в том, что некоторые продавцы всё же предлагают ноутбуки с предустановленным Linux — давайте же посмотрим, что у нас имеется.



► Dell предлагает небольшой, но тщательно подобранный ассортимент ноутбуков с Ubuntu.

**И**так, возможно, стоит начать с вопроса: а почему же нельзя купить Linux-ноутбук? Как мы увидим — можно; разве что не в самом крупном магазине. Вот вам суровая правда: розничные торговцы боятся Linux. До того, как хипстеры начали расхватывать продукты Apple, истина заключалась в том, что всё, на чем работает не Windows, требует дополнительной поддержки, а значит, обходится дороже. Прежде чем торговец затарится чем-то за деньги, он захочет, чтобы кто-то заплатил за это вперед, покрыв ему расходы.

Этим годами занималась Microsoft; они реально платили компаниям кучу денег за закупку своих продуктов или их продвижение в качестве топовых. В мире Linux, по крайней мере, в сфере настольных ПК, нет никого, способного на такое.

Даже если бы производители и хотели создать устройство с Linux, его никто не взялся бы продавать. И опять же, для производителей это та же самая история: создание специфического устройства Linux повлечет за собой дополнительные расходы на поддержку, потенциально — с очень небольшим возвратом средств.

Но это не останавливает некоторых производителей, и они предлагают опции Linux. Мы видели Dell XPS 13 [см. Обзоры, стр. 24 LXF198] и есть линейка Dell Precision, состоящая из M2800, M3800

и M6800. Ни один из них не является обычным ноутбуком: все предназначены для рынка рабочих станций. M2800 стоит от £1000, а два остальных — от £2000 и выше, и это — отличные машины для высокотехнологичной работы.

Есть хорошая новость в виде нового Dell Inspiron 3000 Ubuntu начального уровня, который стоит от £200 и предлагает Ubuntu в качестве опции. С Intel Celeron N3050 и 4 ГБ памяти, мы подозреваем, что это — ответвление линейки Chromebook, и он может быть солидным, хоть и малоинтересным, ноутбуком.

Солидные имена не исчерпываются Dell. HP недавно (примерно в середине 2014 г.) экспериментировала с небольшой линейкой недорогих ноутбуков, которая начинается со скучного, но функционального HP 255 G1 [см. Обзоры, стр. 20 LXF188] с предустановленным Ubuntu 12.04 LTS и разумной системой восстановления. К самому последнему поколению HP 255 G3 присоединились HP 355 G3 и HP 455 G3, с ценой от £199 до £300.

Хотя мы бы не стали с уверенностью рекомендовать базовый компьютер с AMD A4-5000 APU, HP 455 G3 с A10-7300 и 8 ГБ памяти, он вполне может конкурировать с дешевым Intel Core i5 4200U как по скорости обработки, так и по возможностям



► Вы не можете напрямую купить ноутбук Linux Lenovo, но Ubuntu и Lenovo трудятся над сертификацией многих.

их моделях класса рабочей станции Linux предлагается в качестве опции. Ваша следующая опция — выбрать Linux-ноутбук от независимого сборщика систем. Наши читатели из Северной Америки имеют неплохой выбор: например, компании вроде [System76.com](http://System76.com), [puri.sm](http://puri.sm) и [zareason.com](http://zareason.com) поставляют предустановленные ноутбуки с Linux. В Велико-



► HP пробует воду со своими ноутбуками Ubuntu и начала предлагать лучше оснащенный HP 455 G3.

## Ваша следующая опция — выбрать Linux-ноутбук от независимого сборщика систем.

3D-игр. Фактически, на нем можно играть в недавно созданные игры типа *Alien Isolation* (с минимальными настройками), что делает его весьма впечатляющим ноутбуком за свои деньги.

### Независимые и свободные

Невероятно, но это — конец истории Linux для розничных предприятий. Lenovo играет тут отдельную роль, но мы дойдем до них позднее, поскольку они не выпускают ноутбуков с предустановленным Linux; правда, на нескольких

британии имеется <http://minifree.org>, предлагающий интересный, хоть и устаревающий Libreboot Lenovo X200. Он сертифицирован FSF — и использовался никем иным, как самим Ричардом Столлменом [Richard Stallman] — как полностью открытый и свободный, использующий Libreboot UEFI и дистрибутив Trisquel.

Если вы убежденный сторонник свободы ПО, то это — одна из немногих опций на рынке, которую можно купить прямо с прилавка и удовлетворяющая всем запросам. Однако за это вы заплатите ►►

## Chromebook — это же Linux, так?

Признаемся, нам полюбили Google Chromebook [см. Сравнение, стр. 24 LXF202], но мы также отлично понимаем, что вы можете не разделять наших чувств. Мы сами не вполне понимаем, за что мы их так любим; но в основном причина в том, что они просто работают, имеют невысокую цену и на них стоит Linux. Помогает также то, что вы можете использовать *Crouton*, чтобы добавить настоящий GNU/Linux через *chroot* в форме Ubuntu; но может ли это сравниться с установкой по умолчанию?

Итак, на Chromebook работает Linux, и Google чит лицензии с открытым кодом, поэтому поддерживает проект *Chromium OS*. Это означает, что при желании вы можете «собрать» собственный

хромбук с индивидуальной комплектацией [см. Учебники, стр. 64 LXF199]. Хромбуки прошли долгий путь и поставили цель в 2016 г. занять 7% рынка ноутбуков. Они идеальны для работы в Сети, печати, чтения, и умеют воспроизводить фильмы, видео и музыку. То есть достаточно универсальные, в конечном счете.

Система *Crouton* [см. Учебники, стр. 74 LXF204], поддерживаемая инженерами Google, создает *chroot*-установку Ubuntu. Она хорошо работает, но не рассчитывайте на разгон видео на моделях не от Intel. Недавнее расширение *Crouton* даже позволяет Ubuntu работать в окнах вместе с Chrome OS. Поэтому если вам нужен доступ ко всем вашим инструментам GNU/Linux, то он есть, и хромбук

по сравнению с обычным недорогим ноутбуком выигрывает по качеству сборки и дизайну.



► Хотите нечто чуть более гибкое? Хромбук может предложить простоту в использовании и мощь Linux.

производительностью, вплоть до устаревающего, но компетентного (в свое время он был прекрасен) процессора Intel Core 2 Duo P8400.

И даже это еще не открытое оборудование: оно просто отвечает некоторым параметрам свободы ПО. У вас не получится просто пойти и создать собственный Lenovo Libreboot X200, это не свободное оборудование. (Примечание: мы рассмотрим опции открытого оборудования ближе к концу статьи.) Идея создания собственного ПК с нуля весьма непростая, и мы ее рассматривали в **LXF199**. Так что настраивайтесь на эту тему.

## Установите его сами

Возможно, сейчас вы подумаете, что проще установить Linux самому — и будете правы. Сейчас мы потратим некоторое время, чтобы разобраться, как купить готовый ноутбук, на котором точно будет работать дистрибутив GNU/Linux.

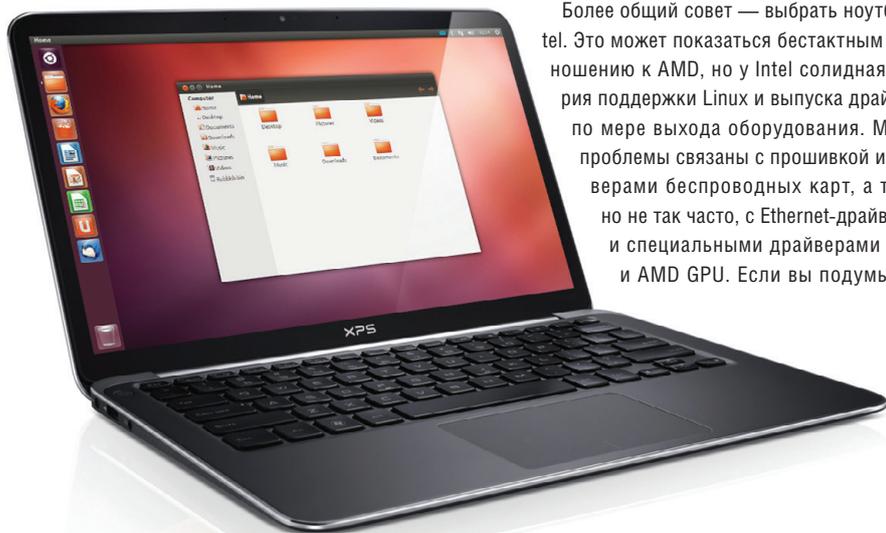
Для начала, Ubuntu ведет список проверенной совместимости не только для ноутбуков, но также и для серверов и настольных ПК — на [www.ubuntu.com/certification/desktop](http://www.ubuntu.com/certification/desktop). Он охватывает целый ряд

производителей, включая Lenovo, Dell, HP и Asus со списком поддержки Ubuntu 12.04 LTS и 14.04 LTS. Посматривая этот список, вы найдете все модели, и это даст вам полный перечень компонентов, BIOS и дополнительные примечания. Это солидный старт, поскольку для Ubuntu 14.04 LTS перечислено более 160 ноутбуков, и более 500 для 12.04 LTS.

Это идеальная ситуация, когда вы знаете, что все компоненты ноутбука поддерживаются Linux. Если вы подумываете о приобретении ноутбука, которого нет в списке Ubuntu, вам придется проверять каждый компонент ноутбука по списку сертифицированных компонентов Ubuntu на [www.ubuntu.com/certification/catalog](http://www.ubuntu.com/certification/catalog); этот список не исчерпывающий, но всё же начало.

Далее вы можете проверить <http://linux-drivers.org>, причем отдельные дистрибутивы предлагают собственные списки поддерживаемого оборудования: например, у Debian имеется база данных (<https://wiki.debian.org/Hardware>), как и у openSUSE (<https://en.opensuse.org/Hardware>) и Linux Mint (<http://community.linuxmint.com/hardware>).

Более общий совет — выбрать ноутбук Intel. Это может показаться бестактным по отношению к AMD, но у Intel солидная история поддержки Linux и выпуска драйверов по мере выхода оборудования. Многие проблемы связаны с прошивкой и драйверами беспроводных карт, а также, но не так часто, с Ethernet-драйверами и специальными драйверами Nvidia и AMD GPU. Если вы подумываете



► Флагманский ноутбук Linux от Dell — работающий на Ubuntu Dell XPS 13 Developer edition.



► Libreboot X200: Возьмите классический Lenovo X200 и установите на нем полностью свободное ПО.

о приобретении ноутбука, постарайтесь удостовериться, что поддержка Linux есть, и по возможности избегайте отдельных GPU.

Последний, но важный момент в том, что некоторые производители ноутбуков блокируют UEFI BIOS, исключая установку другой операционной системы. Трудно сказать, насколько широко распространялась эта тенденция, но мы рекомендуем не покупать ноутбук, не убедившись, что вы сможете загрузиться с другого устройства. Даже если нельзя отключить Secure Boot, вы всё равно должны суметь установить Ubuntu, openSUSE и Fedora, поскольку у них есть ключи благодаря нашему милосердному государю Microsoft. Хм-м, хм... Следует предупредить, что эти ключи не обязательно установлены в UEFI — хотя обычно так и есть; если нет, их можно установить из Windows.

## Открытое оборудование

Люди становятся всё требовательнее, когда дело доходит до закрытых и проприетарных программ и оборудования. Бурный рост встраиваемых систем в форме смартфонов и планшетов выявил, насколько запертым стало оборудование из-за программ загрузки и невидимых прошивок с закрытым кодом. Процессоры настольных ПК тоже настолько усложнились, что на них работают собственные

## Android — не ОС!

Еще одна опция Linux — пойти по пути Android. Несколько смелых компаний попытались протолкнуть ноутбуки Android и гибридные устройства Android, такие как Dell Venue 10 7000 [см. Обзоры, стр. 17 **LXF202**].

Альтернативой этому будет взять планшет Android и выбрать одну из многих клавиатур Bluetooth или просто купить клавиатуру и мышь Bluetooth, поскольку Android через доступное ядро Linux поддерживает координатные устройства и полноценные клавиатуры.

Ограничение, как и с хромбуками, в том, что многие не считают Android полноценным дистрибутивом GNU/Linux, и в этом есть резон. Мы использовали для написания Nexus 5 с отдельной клавиатурой Bluetooth. Это работает, но многозадачность, как

и следовало ожидать, сильно ограничена: например, при переключении между приложениями по Alt+Tab вы увидите, что в половине случаев они закрываются и потом перезапускаются с самого начала. Запуск чего бы то ни было кроме Gmail и Chrome оказывается ограничением.

Лично мы к тому же не думаем, что приложения Android хорошо работают на больших экранах, и нам больше нравится подход Chrome OS — использование браузера, поскольку приложения браузера для таких экранов и разработаны. Однако по большей части вопрос в целевом назначении вашего устройства: планшеты и телефоны Android отлично подходят для работы с низким потреблением ресурсов, но запустите более двух приложений — и они начнут спотыкаться.



► Несмотря на пару неплохих попыток, гибридные ноутбуки Android не радуют из-за плохой реализации многозадачности.

## Рабочие станции Linux

Когда в дверь стучится мир бизнеса, производители реагируют чутко, поскольку тут есть шансы заработать. И если вы ищете высокотехнологичную рабочую станцию, то с ней вам повезет куда больше. Отдельные сегменты мира бизнеса — например, НАСА и нефтехимия — на деле требуют устройств с поддержкой Linux, и у них тугие кошельки. Это, кстати, означает, что такие рабочие станции не из дешевых.

Мы упомянули линейку Dell Precision и рассказали о HP Zbook 15u G2 [см. Обзоры, стр. 20 LXF196], и планируем рассказать о новом Lenovo ThinkPad P70. Это ноутбуки-тяжеловесы, и во многом они являются скорее переносными ПК. Но если вас

интересует возможность обработки большого количества данных на ходу, они более чем способны на это. Нам также следует упомянуть [System76.com](http://System76.com) (разработчик из США): если вы проживаете в США или готовы обойтись без гарантийной поддержки, то их Serval WS и Orynx Pro, например, очень мощные опции.

Упомянутые модели предлагают дисплеи Full HD или даже лучшего качества, процессоры настольного уровня — имеются в виду модели Core i7 и Xeon — 8+ ГБ ОЗУ и специальную графику, например, AMD FireGL или Nvidia Quadro, если она вам нужна для визуализации, плюс мощь OpenCL/CUDA GPGPU.



► На рынке высоких технологий о Linux хорошо позаботились, например, рабочие станции Lenovo P50 и P70.

обновляемые прошивки, и именно их закрытый характер тревожит тех, кто озабочен безопасностью. Кто знает, что творится в самых дальних сумрачных закоулках вашего процессора?

Это обусловило призывы к обеим компаниям открывать код своих самых секретных загрузчиков и прошивок или выпускать оборудование без подобных запертых областей. Как реакция на это, начало появляться открытое оборудование. Хотя раньше создание собственных процессоров было настолько дорогостоящим, что лишь немногие из крупных корпораций могли позволить

себе такое, сейчас есть шансы и у предпринимателей попроче.

Какое отношение это имеет к приобретению ноутбука Linux? Как Trisquel позволяет работать Libreboot X200, так и Linux находится в сердцевине этих новых проектов открытого оборудования. Самым известным является проект Novena, который успешно — на данный момент — запустился на Crowd Supply, собрав более \$700 000 при заявленной цели \$250 000.

ОС основана на Freescale i.MX6 ARM — однокристальной системе (system-on-a-chip — SoC).

В отличие от многих других SoC, она открыта настолько, насколько это возможно. Прошивку нельзя обновлять, и она считается с чипа. Спецификации процессора также доступны без договора о неразглашении. Главным недостатком проекта Novena является отсутствие аппаратного ускорения видео. У Freescale SoC имеется GPU, но, как это часто бывает, код его драйвера закрыт. Надо отдать должное Novena: проект не складывает оружия и идет по пути программного рендеринга. Его надежда и цель заключается в обратном инжиниринге GPU; 2D-ускорение в основном готово, но конечный приз — поддержка 3D OpenGL.

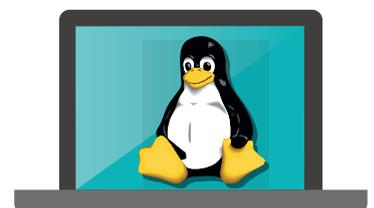
Именно из-за этих компромиссов вы вряд ли выберете Novena: он четко нацелен на фанатов открытого оборудования. Хотя он предлагает четырехъядерный процессор на частоте 1,2 ГГц, применение архитектуры Cortex A9 ARM ограничивает его до 32 бит, поэтому по скорости он вряд ли совершит переворот; но он очень открытый. Еще важнее то, что он указывает путь более мелким производителям SoC, поскольку есть спрос на открытое оборудование. Хотя Intel, возможно, никогда не откроет свой x86, конкуренция на рынке ARM достаточно, чтобы это могло произойти. Это просто вопрос времени.

Итак, отныне вы должны быть полностью вооружены всем, что вам надо знать для успешной покупки солидного ноутбука с Linux: от готовых моделей и сертифицированных предложений Ubuntu до опций самостоятельной установки и открытого оборудования. Совместимость с Linux сейчас превосходная, и если вы приложите небольшие усилия, чтобы избежать системы с заблокированным

UEFI, то можете рассчитывать на счастливую жизнь своего ноутбука. LXF

## Как реакция на это, начало появляться открытое оборудование.

► Novena: самая передовая технология в разработке открытого оборудования, но продукт дорогой, медленный и выглядит по-любительски.





# Tor: Запуск скрытых сервисов

**Джонни Бидвелл не покупается на всю эту риторику «нечего скрывать — нечего бояться», и собирается рассказать вам, как разместить собственный скрытый луковичный сайт.**

**Т**еперь, когда Закон о следственных полномочиях занял свое место в своде законов, в Великобритании утвердились одни из самых строгих надзорных законов в мире [Ред.: — Ага! Великобритания — номер 1!]. Интернет-провайдеры обязаны вести запись истории посещения сайтов клиентами, доступ к которой может быть получен разными организациями.

Применение VPN ширится по мере того, как всё больше людей осознают, что их общение может оказаться объектом необоснованной слежки или храниться без должной защиты, что повышает риск похищения хакерами. Мы уже рассказывали об этом с другой (клиентской) стороны раньше — очень легко настроить роутер Tor с помощью Raspberry Pi [см. Учебники, стр. 72 LXF196], чтобы подключить NetworkManager через коммерческого

провайдера VPN или даже настроить OpenVPN на удаленном сервере. А если вы хотите анонимно запускать сервис — например, web-сервер? Тогда нужна другая стратегия.

Сеть Tor обеспечивает определенную степень анонимности, шифруя данные и направляя соеди-

в центре обработки данных) смогут общаться, не зная друг о друге ничего. Однако Tor делает еще лучше, и настаивает на двух узлах; таким образом, ни один промежуточный узел не знает ни исходной, ни конечной точки, и поэтому нет ни одного «слабого» узла. Можно утверждать, что добавление узлов обеспечит большую безопасность, что весьма вероятно; но заодно и увеличит задержку, которых в Tor и без того хватает, так что это — путь в три стадии. После каждого перепрыгивания удаляется уровень шифрования, что открывает следующий пункт назначения пакета. Этот метод снимает уровни с пакета по очереди, чтобы узнать его следующую остановку, и известен как «луковичная маршрутизация», а сеть Tor (это имя когда-то было акронимом от The Onion Routing — Луковичная Маршрутизация) является хорошо известным ее примером.

**Теперь у Великобритании самые строгие законы по надзору в мире.**

нение через ряд компьютеров и узлов [relay]. Каждый узел может дешифровать только адрес своего следующего соседа и знает только своего предшественника; передавая через них, исходная точка (скажем, компьютер в Великобритании) и конечная точка (скажем, наш web-сервер Tor где-нибудь



Используют Tor двумя разными способами; самый частый — в качестве прокси для обеспечения анонимного доступа к обычным сайтам или другим сервисам. Здесь используется клиентский компьютер, подключенный к каналу Tor, который прерывается на так называемом выходном узле. Выходной узел потом подключается через обычный Интернет (далее именуемый Видимый Интернет) к сайту (или другому сервису), передавая запрос клиента и шифруя ответ перед отправкой его назад по цепи коммуникаций.

Настройка выходного узла создает определенный риск, поскольку в известных обстоятельствах этот пользователь может быть обвинен в действиях тех, кто использует его узел. Теоретически этого не должно случиться, поскольку к сети Tor применяются разные средства обеспечения безопасности, но многие из тех, кто выделяет свой канал, предпочитают предпринять меры предосторожности и выбирают вместо этого настройку т.н. промежуточного узла [middle relay]. Существует также вероятность вредоносного выходного узла, который может пассивно проверять трафик по мере его прохождения, собирая аутентификационную информацию, за которой он шпионит, или активно вмешиваться в трафик, внедряя вредоносный JavaScript или иным путем. Есть вероятность слишком

переоценить эту угрозу — в общем-то, такой риск присутствует при любом доступе к незашифрованным сайтам Видимого Интернета; но и совершенно игнорировать ее тоже не стоит.

## Знай свою луковицу

Можно также разместить сервисы в сети Tor, избегнув проблемы небезопасных выходных узлов. Доступ к таким сервисам (которые в СМИ массово именуются Темной Web, или Даркнет, или иным, столь же глупым образом) можно получить только через сеть Tor. По сути, это не так, поскольку шлюзы вроде Tor2web могут получить доступ к сайтам Tor из Видимого Интернета; но их использование сразу снимает всю защиту анонимности, предлагаемую Tor.

Скрытый сервис Tor можно запускать откуда угодно; в отличие от многих других приложений, здесь нет проблем с обходом NAT, поэтому незачем переадресовывать произвольные диапазоны портов, чтобы заставить его работать из вашей домашней сети. Для всего, кроме частных проектов, советуем запускать сервис на отдельном компьютере, чтобы ничем не рисковать, если компьютер будет взломан. Однако если вы планируете обслуживать большой объем трафика, то рано или поздно вам придется потратиться на хостинг с соответствующими

```
DataDirectory /var/lib/tor
HiddenServiceDir /var/lib/tor/
hidden_service/
HiddenServicePort 80 127.0.0.1:80
```

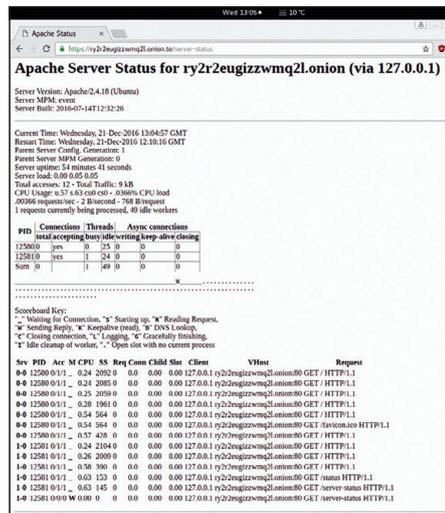
Они сообщат Tor, где находить его данные времени исполнения, где хранить данные о нашем скрытом сервисе, плюс слушать только интерфейс обратной петли. Это уже шаг к неслучайной деанонимизации, еще до того, как мы вообще начали. С нашей новой настройкой запустите (или перезапустите, если он уже запущен) сервис Tor с помощью

```
$ sudo systemctl start tor
```

Если всё прошло по плану, должна создасться директория `/var/lib/tor/hidden_service`, содержащая два файла: имя хоста нашего скрытого сервиса и частный ключ. Давайте выясним первое, чтобы знать, где искать наш скрытый сервис:

```
$ sudo cat /var/lib/tor/hidden_service/Hostname
```

Если эта директория или файл не создались, проверьте логи с помощью `journalctl -u tor` и выясните, что случилось. (Небось установили пакеты прямо из репозитория Ubuntu? А ведь мы вам ска-



» Модуль состояния Apache бывает очень полезен при диагностике проблем с вашим сайтом; но, к сожалению, не столь полезен для сохранения анонимности вашего сайта или ваших посетителей.

## Установка Tor

Tor можно найти в репозиториях всех основных дистрибутивов. Пакет в репозитории Debian не будет последней версией, но отладки безопасности бэк-портируются в него весьма быстро, поэтому для всех, кроме самых параноидальных, он вполне подойдет. Злостные параноики, возможно, предпочтут добавить собственные репозитории проекта Tor, следуя инструкциям с <https://www.torproject.org/docs/debian.html.en>. Однако указания от проекта

## Можно разместить сервисы в сети Tor, избегнув проблемы небезопасных выходных узлов.

Трафиком от абонента к коммутирующему узлу.

Следуйте инструкциям внизу, чтобы установить Tor в Debian или Ubuntu; в других дистрибутивах он должен быть доступен прямо из менеджера пакетов с минимумом усилий. Хорошо изучите файл настройки Tor — `/etc/tor/torrc`: там много всякой ерунды, но применительно к тому, что мы собираемся делать. Мы сделаем резервную копию с помощью отличного расширения и создадим настройку поменьше:

```
$ cd /etc/ tor
```

```
$ sudo mv torrc{,orig}
```

```
$ sudo nano torrc
```

Вам надо будет заполнить этот файл следующими директивами:

Tor недвусмысленно рекомендуют не использовать пакеты из репозитория Ubuntu, и если вы используете версию 16.04 шедевра от Canonical, создайте файл `/etc/apt/sources.list.d/tor.list` со следующим содержанием:

```
deb http://deb.torproject.org/torproject.org xenial main
deb-src http://deb.torproject.org/torproject.org xenial main
```

Если вы работаете в 16.10, замените `xenial` на `yakkety`. Затем возьмите ключ GPG проекта Tor

(его можно скопировать и вставить из раздела Ubuntu и Debian на сайте) —

```
$ gpg --keyserver keys.gnupg.net --recv
A3C4F0F979CAA22CDBA8F512EE8\CBC9E886DDD89
```

и добавьте его на связку `Apt`:

```
$ gpg --export A3C4F0F979CAA22CDBA8F512EE8\CBC9E886DDD89 | sudo apt-key add --
```

Затем обновите список пакетов по `$ sudo apt-get update` и установите Tor и связку ключей:

```
$ sudo apt-get install tor deb.torproject.orgkeyring
```

[facebookcorewwi.onion](http://facebookcorewwi.onion) в 2014 г. Последние восемь символов были чистой удачей. Вы можете прочитать больше о том, как генерировать ключи и имена хоста, в блоге Тима Тауберта [Tim Taubert] на <http://bit.ly/TorGenKeysAndHostnames>. Директория скрытых сервисов видима только для пользователя **debian-tor**, под которым работает сервис (поэтому автозаполнение с клавишей Tab не будет работать для команд `sudo` выше). Как и с ключами SSH, менять разрешения на эти файлы — плохая идея. А теперь давайте настроим наш web-сервер.

## Установка Apache

Для обслуживания статической web-страницы мы воспользуемся *Apache*, но процесс практически такой же для *Nginx*. Вообще-то некоторые утверждают, что он проще, безопаснее и во всех смыслах лучше для работы с сервисом Tor, но подобные вещи всегда субъективны, а нам нравится *Apache*. Мы пропустим настройку, потому что она весьма обширная [см. Учебники **LXF197**, там вы найдете более глубокое руководство по настройке. Установка на Debian/Ubuntu проста]:

```
$ sudo apt install apache2
```

Воспользуемся модульной структурой сайта Debian, чтобы всё у нас было опрятно и аккуратно и *Apache* мог обслуживать другие сайты. При этом

идея одновременно обслуживать Tor и сайты Видимого Интернета, видимо, плохая, поскольку умный и целеустремленный злоумышленник, сопоставив цифровые отпечатки сервера для них обоих, поймет, что они идут из одного источника.

```
$ cd /etc/apache2/sites_available/
$ sudo cp 000-default.conf lxftor.conf
$ sudo nano lxftor.conf
```

Замените `*:80` в первой строке *lxftor.conf*, чтобы она выглядела так:

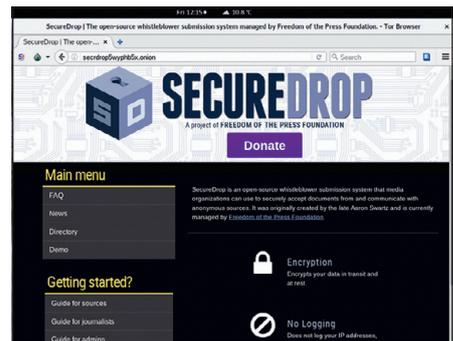
```
<VirtualHost 127.0.0.1:80>
```

Это велит *Apache* обслуживать наш сайт только в локальном интерфейсе, чтобы запросы по его открытому IP-адресу обслуживали настройки по умолчанию (пустой сайт). Запросы через Tor перенаправляются клиентом Tor в интерфейс обратной петли, чтобы они не игнорировались. Также прокомментируйте строку `ServerName` и укажите в ней свой `.onion`-адрес; плюс замените строку, начинающуюся с `DocumentRoot`, на

```
DocumentRoot /var/www/tor
```

Здесь будет храниться наш HTML. Сохраните этот файл, создайте корневой документ для нашего сайта Tor и заполните его каким-нибудь контентом:

```
$ sudo mkdir /var/www/tor
$ echo Hello Tor world! > sudo tee /var/www/tor/index.html
```



SecureDrop предлагает сайт `.onion`, где правдоискатели и журналисты могут спокойно общаться (используя GPG).

Теперь надо включить наш сайт Tor:

```
$ sudo a2ensite lxftor
```

В принципе, наличие сайта «за всё» по умолчанию — идея хорошая, поскольку это означает, что в результате неправильной настройки будет обслуживаться по большей части безвредный сайт, но не ваш скрытый сервис. Многие предпочли бы, чтобы *Apache* вообще не слушал внешних адресов, и это очень легко реализовать. Мы можем сделать это с помощью брандмауэра или просто отредактировать `etc/apache2/ports.conf`, заменив строку `Listen 80` на `Listen 127.0.0.1:80`.

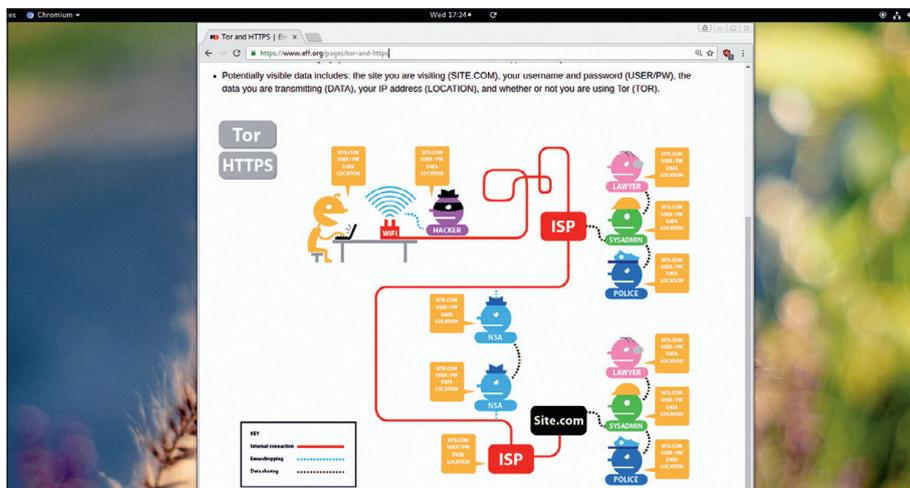
Обратите внимание, что подключения при этом активно отклоняются, а не тихо сливаются, так что наблюдатель может сообразить, что там присутствует сервис. Будет ли это лучше, хуже или так же, как обслуживать пустой сайт, зависит от ряда других факторов. Пока вы не приняли окончательного решения, советуем не редактировать `ports.conf`. Теперь перезапустите сервис *Apache* (технически здесь должно хватить `reload`, но иногда таким образом пропадают изменения в `config`), и мы должны обслуживать контент в Tor:

```
$ sudo systemctl restart apache2
```

Сначала проверьте, чтобы наш сайт работал по адресу заковычкивания:

```
$ wget -qO- localhost # Это заглавная O, а не ноль!
Hello Tor world!
```

Если сообщение `hello world` [привет, мир] не отображается, значит, что-то неправильно с вашей



EFF создали альтернативную страницу (<http://bit.ly/EFFTorAndHTTPS>), показывающую, от каких угроз защищает Tor.

## Tor Browser Bundle

Даже скромному web-браузеру доверять нельзя, ибо он может выдать инкриминирующие подробности, вот почему Tor Project постоянно говорит о необходимости использовать Tor Browser Bundle, а не просто защитить выбранный вами браузер с помощью прокси. *Tor Browser* создан на ESR (Extended Support Release — релиз с расширенной поддержкой) *Firefox*, поэтому в нем нет самых свежайших функций, но зато он получает достаточно пристальное внимание и со стороны Mozilla, и со стороны Tor Project. Он также очень старается обеспечить безопасность подключения к Tor и может сгладить проблемы с NAT, прокси или брандмауэрами.

Лучший способ скачать *Tor Browser* — прямо с [www.torproject.org](http://www.torproject.org) (мы не будем указывать ссылку [bit.ly](http://bit.ly), и предоставляем вам перейти в нужный раздел сайта самостоятельно). Неплохо будет также проверить подписи.

Сначала скачайте браузер и соответствующий файл `.sig` с сайта Tor Project. Затем импортируйте ключ Tor Project с помощью

```
$ gpg --keyserver pool.sks-keyservers.net --recv-keys 0x4E2C6E8793298290
```

Теперь проверьте, что этот ключ действительно принадлежит Tor Project:

```
$ gpg --fingerprint 0x4E2C6E8793298290
```

И, наконец, проверьте подпись (запустите эту команду из директории, где находятся файлы):

```
$ gpg --verify tor-browser-linux64-6.0.8_en-US.tar.xz.asc,
```

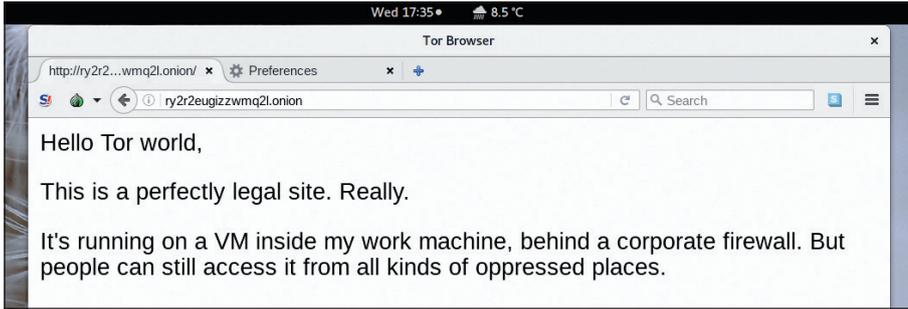
Будем надеяться, нас встретит сообщение «правильная подпись [good signature]». Если нет, то, вероятно, в скачивание кто-то вмешался, и с этой подписью лучше не связываться. В противном случае распакуйте ее с помощью

```
$ tar xvJf tor-browser-linux64-6.0.8_en-US.tar.xz
```

и запустите, с помощью

```
$ cd tor-browser_en-US
```

```
$ ./start-tor-browser.desktop
```



» Как минимум, наш скрытый сервис пресек брандмауэры издательства. Доказав, что незачем запускать базар психонавтов темной сети ради проблем в темной сети.

настройкой *Apache*; попробуйте запустить `apachectl configtest`, чтобы выявить проблему. Если такое сообщение появилось, то следующим шагом будет проверить, что этот самый контент обслуживался через Tor. Мы не настроили наш компьютер на доступ к onion-адресам, поэтому простая вставка содержимого `/var/lib/tor/hidden_service/hostname` в браузер не сработает. Зато сработает использование прокси Tor2web, достигаемое просто добав-

Интернета модуль так себя не ведет — или ведет, но только для подключений в интерфейсе обратной петли (загляните в файл `status.conf`, чтобы посмотреть, как). Проблема в том, что подключения Tor, оказывается, идут из адреса обратной петли, поэтому `mod_status` полагает, что делиться с ними информацией о статусе — это нормально. На самом деле — нет, но это легко исправимо: просто отключите модуль и затем перезагрузите сервис *Apache*.

## Благодаря Let's Encrypt все больше сайтов обслуживают контент, зашифрованный через HTTPS.

лением `_to` (или `_ink`, или `_cab`, или `_pu`, опций много) к onion-URL. Если вас терзает паранойя (ну и в зависимости от сервиса, который вы в итоге планируете запускать), то у вас, возможно, есть сомнения в таком образе действий. Во-первых, это открывает ваш луковичный адрес всем, кто следит за вашей деятельностью в Интернете. Большинство людей на самом деле даже не считают нужным скрывать эту информацию, но уж лучше всё делать как полагается. А проверять работу вашей маршрутизации `.onion` полагается попыткой получить доступ к вашему сайту с другого компьютера, где есть Tor Browser Bundle (см. врезку внизу стр. 52), или, возможно, с устройства Android, на котором работает *Orbot* (тоже созданный проектом Tor). Если всё работает с адресом `localhost`, но не работает здесь, то в настройке Tor что-то не так. Помочь может перезапуск сервиса Tor.

```
$ sudo a2dismod status
$ sudo systemctl reload apache
```

И теперь мы можем спокойно игнорировать все эти истеричные заявления в СМИ от февраля 2016 г. Благодаря Let's Encrypt всё больше и больше сайтов обслуживают контент, зашифрованный через HTTPS. Во многих областях применения аутентификационная часть HTTPS (когда вы скачиваете сертификат, подписанный организацией, которой вы должны доверять) не имеет особого смысла. Если вы пытаетесь скрыть местоположение своего сайта, то вам труднее будет убедить уважаемый орган сертификации выдать требуемую подпись.

Также нет стандартов сертификации доменов `.onion`, хотя эта работа ведется. Бывают случаи, когда работа «скрытого сервиса» не подразумева-

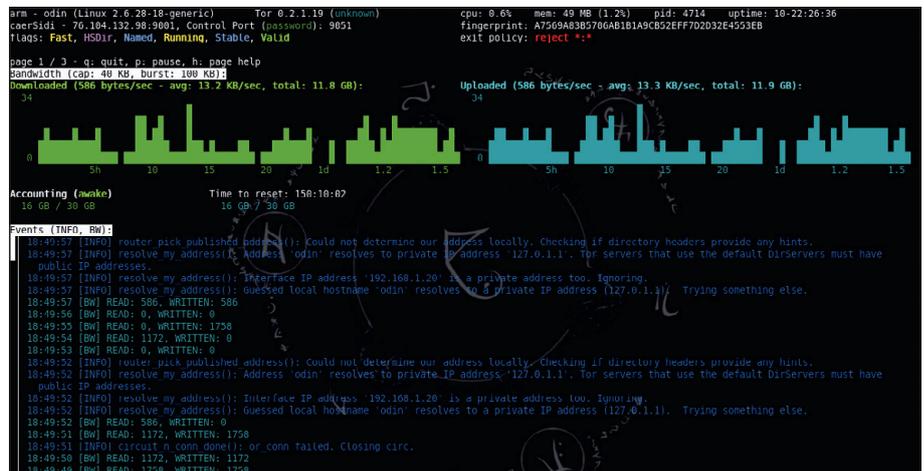
ет, что его адрес должен быть скрыт от публики. Яркий пример подобного — упомянутый выше случай с Facebook. Этот сайт существует не как некая тайная соцсеть, работающая только по приглашению, а чтобы люди получали доступ к Facebook, не опасаясь вмешательства со стороны правительственных органов или хакеров. В таких случаях HTTPS определенно окажется логичнее, и ведется работа по внедрению этой технологии в `.onions`. Другая часть HTTPS, шифрование, никак не вредит Tor, и если вы хотите создать и самостоятельно подписать собственный сертификат, то процесс будет таким же, как и для обычных сайтов. Однако добавление еще одного уровня шифрования потенциально тоже излишне, потому что луковичная маршрутизация по определению в любом случае употребляет несколько уровней шифрования для всего. Всё это применимо к доступу к `.onion`-сайтам, но для доступа к сайтам Видимого Интернета через Tor доверяйте HTTPS. Он затруднит коварным выходным узлам провайдеру скрытого сервиса и также узла Tor. Но это плохая идея, потому что узлы должны быть публично перечислены — чтобы любой, у кого есть время, смог соотносить трафик между узлом и севером. Если что-то пошло не так и ваш хост упал, перехватчик заметит, что нет ни узла, ни сервера, и сумеет сделать вывод, что на самом деле это один и тот же компьютер. Кроме того, запуск узла Tor требует, чтобы порты были доступны для внешнего мира, и нет такой магии, которая заставила бы это работать изнутри роутера.

Проблема видимости узлов частично была решена благодаря мостовым узлам, настраиваемым из *Tor browser*. Это означает, что если даже правительство заблокирует доступ к открыто перечисленным узлам Tor, как это недавно случилось в Турции, пользователи всё равно смогут подключаться. **LXF**

Проблема видимости узлов частично была решена благодаря мостовым узлам, настраиваемым из *Tor browser*. Это означает, что если даже правительство заблокирует доступ к открыто перечисленным узлам Tor, как это недавно случилось в Турции, пользователи всё равно смогут подключаться. **LXF**

## Прячем свой сервер

Для тех, у кого работают скрытые сервисы, настройка *Apache* по умолчанию в Debian и Ubuntu проблематична. Главная проблема тут в том, что модуль `mod_status` по умолчанию включен, и это может привести к утечке самой разной информации: достаточно добавить `/serverstatus` к адресу `.onion` (попробуйте — и содрогнитесь). Выяснится не только какая версия *Apache* запущена, но и состояние загрузки сервера, и — что, возможно, самое ужасное — все-все подробности недавних запросов GET. А так как всё это должно появляться из адреса обратной петли, оно не то что выдаст IP-адрес посетителя, но можно заполучить любую другую информацию (в зависимости от структуры вашего сайта), изучив URL. Для сайтов Внешнего



» *Arm* (Anonymizing Relay Monitor) — инструмент на базе *ncurses* для визуализации активности Tor.

# 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](http://www.linuxcenter.ru)

Linux  center

Наши эксперты помогут вам с любым приложением Linux!



**ЕВГЕНИЙ БАЛДИН**  
Подтвердивший  
свою квалификацию  
физик.

## Kickstarter и Git

Есикава: Скажите, почему вы решили стать учителем?

Онидзука: Если нужна причина — ну, по приколу.

*Крутой учитель Онидзука, Эпизод 2.*

**С**вободный софт для разработчиков на Kickstarter — это как минимум странно.

Да, дважды сообществу Krita Foundation удалось собрать по €30 и 40 тыс. в 2015 и 2016 г. соответственно. Художники тоже дело делают. И была эпическая история с собранными \$300+ тыс. за обещания сделать текстовый супер-пупер-редактор для разработчика *Light Table*, в итоге оказавшаяся разводкой: ситуация крайне печальная, но, с другой стороны — вложившиеся должны понимать, что лучший текстовый редактор уже есть, и имя ему — *Emacs*!

Джонас Бернулли [Jonas Vernoulli] предложил скинуться себе на годовую зарплату. Он хочет не так уж много — 55 тыс. швейцарских франков. За что? — может спросить российский читатель. А за то, что Джонас — ключевой человек в поддержке и развитии *magit*, интерфейса для работы с *Git* из *Emacs*. С его слов — это “killer app” для *Emacs*, наравне с *org-mode* (если вы не знаете, что это такое, срочно исправляйте данное упущение). Довольно наглое утверждение; но есть те, кто вправду так считают, и Джонас явно получит запрошенное.

**PS** Есть софт — есть и баги; но набор уязвимостей *BlueVogne* может реально убить *Bluetooth* у устройств, не получающих обновлений. Я, к сожалению, на *Android*-телефоны намекаю, большинство из них.

[E.m.Baldin@inp.nsk.su](mailto:E.m.Baldin@inp.nsk.su)



## В этом месяце вы научитесь...

### Шутить в терминале ..... 56 Следить за процессами ..... 58

Если всё время вкалывать, не оставляя ни часа на потеху, легко стать занудой. **Ника Пирса** такая перспектива не устраивает.

**Дмитрий Пантелеичев** наблюдает за процессами и управляет ими с помощью пакета системных утилит нового поколения.

### Объединять файловые системы ..... 62 Спасать данные ..... 64

**Джонни Бидвелл** карабкается на виртуальное дерево, преодолевая уровни абстракции...

Ассистируя доктору **Туксу**, **Маянк Шарма** восстанавливает утраченные было файлы с практически хирургической точностью.

### Шифровать файлы ..... 68 Эмитировать валюту ..... 72

**Нейт Дрейк** с удовлетворением обнаружил, что в *Ubuntu Linux* программы шифрования давно уже предустановлены.

Поучитесь у **Нейта Дрейка** совершать революцию в отдельно взятой экономике — тому есть исторические примеры.

### Ставить ZFS ..... 76 Тушить свет ..... 80

Вслед за **Михалисом Цукалосом** установите себе журналируемую файловую систему с разделами *RAID*. Это если матчасть позволяет.

Кто добрался до лампы в *Послушном Доме* и погасил ее? **Максиму Черепанову** такое не до лампочки, он во всем разберется.

## АКАДЕМИЯ КОДИНГА



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

### И снова о числе π ..... 84 Копии на Python ..... 88

Иррациональное число  $\pi$  влечет не одного **Михалиса Цукалоса**. Множество математиков пытались его расчитать с заданной точностью; попробуем и мы.

**Михалис Цукалос** проводит очередную физзарядку мозгов: выполняем упражнение по копированию файлов и каталогов.

# Terminal: Время посмеяться

Ник Пирс решил на время отставить серьезность и исследовать шутивную сторону *Terminal*, чтобы раскрыть некоторые забавные эффекты.



Наш эксперт

Ник Пирс в этом месяце вдоволь наигрался с ASCII и подурчался с помощью терминала. Неплохо для разнообразия.

До сих пор в нашей постоянной серии, посвященной терминалу, мы смотрели на вещи с точки зрения их практической пользы. Но ведь в жизни есть место не только серьезности, так что давайте пройдемся и по шутивным командам и инструментам, которые вы можете добавить в свой репертуар.

## Опечатки — это весело

Приходилось случайно набрать `sl` вместо `ls`? Казалось бы, система должна выдать ошибку «Команда не найдена», но Ubuntu вместо этого предложит вам установить загадочный инструмент `sl`. `Sl` означает Steam Locomotive — но чем рассказывать, лучше сами в следующий раз ошибитесь с `ls` (или наберите `LS`), и полюбуйтесь, как по вашему терминалу проедется паровоз:

```
$ sudo apt-get install sl
```

Среди прочих команд, производящих забавные эффекты — `matrix` (создает копию рабочего стола Хакера из фильма «Матрица [The Matrix]»); `aafire` (`sudo apt-get install libaa-bin`) и `bb` (демонстрация ASCII с дополнительным музыкальным сопровождением).

Мечтаете просматривать любые фотографии или изображения в стилистике ASCII? Наберите `sudo apt-get install aview`, а чтобы всё отображалось в неподражаемом ASCII, введите

```
$ asciiview Path/filename.png
```

Если вы получили ошибку “invalid magic-not”, то надо будет сначала конвертировать файл (при наличии установленного `imagemagick`):

```
$ convert filename.png filename.pgm
```

## Создаем текстовые баннеры

Еще один способ оживить текстовый вывод — выполнить его в виде текстового баннера, и для этого есть целый ряд инструментов, предлагающих немного разные варианты. Начнем с `FIGlet` (`sudo apt-get install figlet`). Базовый синтаксис таков:

```
$ figlet Linux Format Rules!
```

`FIGlet` также может создавать баннеры в разных стилях. Узнайте поддерживаемые опции с помощью команды `showfigfonts`, а затем используйте один из них:

```
$ figlet -f smslant I love Linux!
```

Также стоит обратить внимание на флаги `-t`, указывающий `FIGlet` использовать всю ширину окна терминала (чтобы указать конкретную ширину столбца, воспользуйтесь `-w 70`), и `-p` или `-n`, которые пригодятся, если вы добавляете в `FIGlet` много текста. Главное, выберите `-p` для перевода `FIGlet` в режим абзаца, чтобы он устранил разрывы строк.

Если вы хотите сделать баннеры цветными, установите `Toilet` (`sudo apt-get install toilet`). Принцип работы здесь тот же, что и в `FIGlet`, но помимо разных шрифтов (он поддерживает те же шрифты, что и `FIGlet`), вы также можете указать один из двух цветных фильтров с флагом `-F`: радужный (`-F gay`) или металл (`-F metal`).

```
$ toilet -f smblock -F metal I love Linux!
```

Эти инструменты могут иметь и практическое применение — скажем, в составе скрипта или для того, чтобы разнообразить



» Заголовок в виде баннера в терминале может быть очень даже полезен, например, чтобы выделить часть скрипта.

## Поговорить всегда приятно

Зачем выискивать взглядом баннеры на экране, когда вы можете заставить свой компьютер их озвучить?

`eSpeak` (<https://espeak.sourceforge.net>) — это крошечный речевой синтезатор, доступный в основных репозиториях (`sudo apt-get install espeak`). Базовый его синтаксис таков:

```
$ espeak "The Terminal rules"
```

Слишком тихо? Используйте флаг `-a`, чтобы изменить уровень громкости от 0 до 200 (по умолчанию 100). Вы можете настроить высоту голоса с помощью флага `-p` (0–99, значение по умолчанию 50) и ускорить воспроизведение с помощью `-s` (измеряется в словах в минуту: 175 по умолчанию, с нижним пределом 80 и без верхнего предела, но около 260 — самое оно).

Голос воспроизведения выбран по умолчанию, но его можно сменить. Для начала введите `espeak --voices` — узнаете, что доступно; затем используйте флаг `-v` ТипГолоса для установки выбранного вами языка и акцента. Объедините их, чтобы получилось нечто вроде

```
$ espeak -p 99 -s 200 -v en-scottish "The engines canna take it capt'n!"
```

## Смотреть фильмы в ASCII. А почему бы и нет?

Самый крутой трюк со стилем ASCII — использовать его для просмотра видео, и это стало возможным благодаря комбинации приложения *mplayer* (`sudo apt-get install mplayer`) и *libaca-bin* (см. основной текст) для монохромных рендерингов и *libcasa* (установленным параллельно с *mplayer*) для цветных.

Если ваше видео живет в папке `/Videos`, используйте следующую команду, заменив `/Videos/file.mp4` на путь и имя вашего файла:

```
$ mplayer -vo aa /Videos/file.mp4
```

Во время воспроизведения вы можете изменять контраст с помощью клавиш 1 и 2, а с помощью 3 и 4 можно управлять яркостью. Нажмите 5, чтобы включить и выключить быстрый рендеринг, а 6 — чтобы

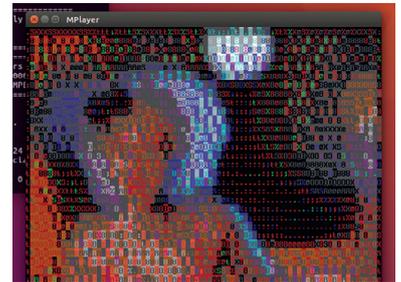
переключать режим принудительных флуктуаций между нулевым, с распределением ошибок и по алгоритму Флойда–Стейнберга [Floyd–Steinberg]. Наконец, для инвертирования изображения нажимайте 7.

Чтобы приостановить воспроизведение, нажмите пробел, а чтобы остановить — Enter.

Если вы хотите смотреть видео в формате ASCII в цвете, замените `aa` на `casa`:

```
$ mplayer -vo casa Videos/file.mp4
```

Библиотека *Casa* хорошо справляется с сохранением пропорций, и вы можете настраивать режимы фона, защиты от наложения спектров и сглаживания, используя во время воспроизведения клавиши B, A и D соответственно.



➤ Плагину *libcasa* неплохо удается воспроизведение цветных фильмов в узнаваемой стилистике ASCII.

представление информации. Например, следующая команда отображает календарь и часы реального времени (для выхода нажимайте Ctrl+C):

```
$ watch -n1 "date '+%D%n%T' | figlet -k"
```

### Плюс текстовые эффекты

Среди других подобных текстовых эффектов — *cowsay*, который просто повторяет любую заданную вами фразу (*cowsay* Я люблю Linux Format), добавляя к ней изображение коровы в виде ASCII-символов — чтобы получить то же в более приятном графическом виде, установите вместо этого *xcowsay*.

Если вы не любитель коров, то вас порадует, что флаг `-f` позволяет выбирать из множества различных форм искусства — от других животных до звезд телевидения и кино:

```
$ cowsay -f hellokitty Linux - это так круто!
```

Есть куча других вариантов, в основном шуточных; все они описаны в `man cowsay`. Стоит упомянуть `-b` для состояния киборга [Borg], `-d` — для мертвеца и `-r` для паранойи. Вы даже можете переключиться в режим задумчивости, используя *cowthink*.

Другие инструменты производят разные текстовые ответы: *fortune* (`sudo apt-get install fortune`) генерирует случайные советы или цитаты типа «Вам не хватает времени; и, возможно, всегда будет не хватать». Установив *rig* (`sudo apt-get install rig`), вы будете получать случайные адреса или телефонные номера США (используйте флаги `-f` и `-m` для определения женских или мужских имен).

Хотите, чтобы текст выводился на экране, как будто он вводится в режиме реального времени? Установите *pv* (`sudo apt-get install pv`) и используйте следующий синтаксис:

```
$ echo "Incoming message from Linux Format: don't forget to renew your subscription" | pv -qL 10
```

Разумеется, вы можете запустить что-нибудь, а затем ускорить набор, увеличив значение `-qL` до, скажем, 20 (или, естественно, уменьшить значение для замедления).

### Установка не требуется

Вы обнаружите, что в Linux полно «пасхалок»-киндерсюрпризов, и многие из них находятся внутри системных команд. Введите `apt-get moo` и нажимайте Enter — вы увидите корову (другую), которая спросит вас, «мычали» ли вы сегодня.

Теперь попробуйте ввести `aptitude moo` — последует короткий ответ, что доступных «пасхалок» нет, поэтому попробуйте `aptitude -v` и обратите внимание на ответ. Пробуйте `aptitude -vv`, `aptitude -vvv` и т.д., пока система терпит. Не поняли, что у вас получилось? Введите `aptitude -vvvvvv moo`, и узнаете.

Команда *rev*, также встроенная в Ubuntu, просто выведет текст задом наперед. Напечатайте `rev`, чтобы ее запустить, затем введите

текст и посмотрите, что она выдаст. Когда вам надоест добавлять фразы, нажмите Ctrl+C для выхода.

Существует некая, но с субтитрами, ASCII-версия Star Wars, доступная онлайн, которую можно посмотреть в окне терминала, используя *telnet*, вот так:

```
$ telnet towel.blinkenlights.nl
```

Наконец, с помощью команды *calendar* в *Terminal* можно просматривать не только базовые календари, например:

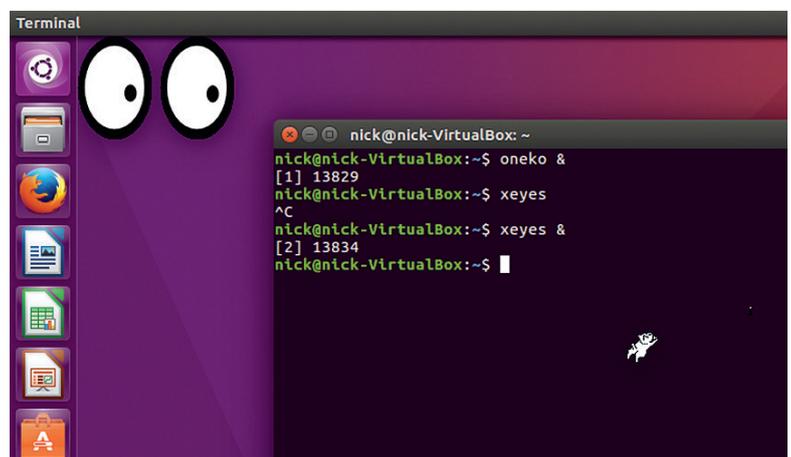
```
$ calendar -f /usr/share/calendar/calendar.computer -A 365
```

В этом списке представлены любопытные существовавшие или придуманные системы летоисчисления. Замените *computer* на *lotr* (Lord of the Rings [Властелин колец]), *pagan* [языческий], *history* [исторический] и *music* [музыкальный] для просмотра альтернативных календарей, или введите `ls /usr/share/calendar/`, чтобы увидеть полный список доступных.

### Одним щелчком мыши

Если вы хотите, чтобы за движением курсора по экрану наблюдала пара глаз, попробуйте запустить *xeyes*, это часть пакета *x11-apps* (при необходимости, `sudo apt-get install x11-apps`). Перед вами появятся два глаза; сдвиньте мышью, чтобы увидеть, как они следуют за указателем. Нажимайте Ctrl+C, чтобы выйти.

Введите `sudo apt-get install oneko`, затем `oneko`, и вы увидите, что на экране появилась милая маленькая кошечка. (Предпочитаете собак? Поставьте вместо этого `oneko -dog`.) Кошка будет преследовать указатель курсора по экрану. Нажмите Ctrl+C, чтобы прекратить этот эффект. LXF



➤ Если вы хотите вызвать определенные команды, не перегружая терминал, просто добавьте `&` в конце команды.

### Скорая помощь

Отругайте любого, кто вводит неверный пароль, покушаясь на *sudo*: введите `sudo visudo` и добавьте верхнюю строку, отмеченную как *Default insults* [Брань по умолчанию]. Сохраните ее и выйдите, затем введите `sudo -k`, а потом `sudo ls`. Введите неверный пароль — и получите в ответ случайное ругательство.

ЧАСТЬ 1

# Procps-ng: GNU/Linux и процессы

Дмитрий Пантелеичев обратился к одной из важнейших тем, касающихся ОС.



Наш эксперт

Дмитрий Пантелеичев считает, что любую технологию надо изучать от простого к сложному, и каждый шаг закреплять практически на примерах.

Главная миссия любой операционной системы — обеспечить выполнение поставленных задач, рационально распределить аппаратные ресурсы. Процесс — это выполняющийся набор команд, которому для работы предоставлена область оперативной памяти и другие необходимые ресурсы, например, файлы. Процесс порождает один или несколько потоков, каждому из которых предоставляется еще один ресурс: процессорное время. Операционная система должна обеспечить изоляцию процессов, чтобы они не вмешивались в работу друг друга.

Пакет *procps-ng* содержит программы, позволяющие наблюдать за процессами и управлять ими. Этот пакет появился как ответвление от старого пакета *procps*. Группа разработчиков Debian, Fedora и openSUSE посчитала, что старый пакет перестал поддерживаться в должной степени, оброс патчами, специфическими для разных дистрибутивов, и потерял универсальность. Поэтому они решили сделать ветку от *procps*, и назвали ее *procps-ng* [next generation — новое поколение].

В пакет входят следующие программы: *ps*, *top*, *pidof*, *pwdx*, *w*, *uptime*, *mpstat*, *pgrep*, *pskill*, *free*, *slabtop*, *sysctl*, *tload*, *vmstat*, *watch*, *skill*, *snice*.

## Программа ps

Начнем с программы *ps*. Ее задача — отображать активные процессы, выбранные по тем или иным условиям. При ее вызове можно вводить параметры, уточняющие условия выбора.

Эти параметры бывают трех типов.

- » UNIX-параметры: однобуквенные, начинающиеся с дефиса;
- » BSD-параметры: однобуквенные, дефиса не имеющие;
- » GNU-параметры: многобуквенные, начинающиеся с двух дефисов.

» Рис. 1. Информация о процессах, полученная через *ps aux*.

```
dima@linux-dmkz: ~
dima@linux-dmkz:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  33684  5820 ?        Ss   15:39   0:01 /usr/lib/systemd/systemd
root         2  0.0  0.0      0     0 ?        S    15:39   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    15:39   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S<   15:39   0:00 [kworker/0:0H]
root         7  0.0  0.0      0     0 ?        S    15:39   0:01 [rcu_preempt]
root         8  0.0  0.0      0     0 ?        S    15:39   0:00 [rcu_sched]
root         9  0.0  0.0      0     0 ?        S    15:39   0:00 [rcu_bh]
root        10  0.0  0.0      0     0 ?        S    15:39   0:00 [rcuop/0]
root        11  0.0  0.0      0     0 ?        S    15:39   0:00 [rcuos/0]
root        12  0.0  0.0      0     0 ?        S    15:39   0:00 [rcuob/0]
root        13  0.0  0.0      0     0 ?        S    15:39   0:00 [migration/0]
root        14  0.0  0.0      0     0 ?        S    15:39   0:00 [watchdog/0]
root        15  0.0  0.0      0     0 ?        S    15:39   0:00 [watchdog/1]
root        16  0.0  0.0      0     0 ?        S    15:39   0:00 [migration/1]
root        17  0.0  0.0      0     0 ?        S    15:39   0:00 [ksoftirqd/1]
root        19  0.0  0.0      0     0 ?        S<   15:39   0:00 [kworker/1:0H]
root        20  0.0  0.0      0     0 ?        S    15:39   0:00 [rcuop/1]
root        21  0.0  0.0      0     0 ?        S    15:39   0:00 [rcuos/1]
root        22  0.0  0.0      0     0 ?        S    15:39   0:00 [rcuob/1]
root        23  0.0  0.0      0     0 ?        S    15:39   0:00 [watchdog/2]
root        24  0.0  0.0      0     0 ?        S    15:39   0:00 [migration/2]
```

Вызов *ps* без параметров выдаст только процессы, запущенные текущим пользователем через текущий терминал:

PID	TTY	TIME	CMD
2440	pts/0	00:00:00	bash
2467	pts/0	00:00:00	ps

Но чаще всего программу *ps* вызывают с параметром *aux*, который является группой из трех BSD-параметров: *a* (отмена ограничения «только я»: показывать процессы, запущенные всеми пользователями), *u* (отмена ограничения «текущий терминал», и увеличение количества столбцов, то есть более расширенная подача информации), *x* (отмена ограничения «только терминал»: показывать и процессы, не запущенные через эмуляторы терминала): *ps aux* (рис. 1).

Получившийся результат состоит из следующих столбцов: *USER* (имя пользователя), *PID* (идентификатор процесса), *%CPU* (время работы процессора в процентах), *%MEM* (отношение физической памяти, предоставленной процессу, к общему количеству физической памяти в процентах), *VSZ* (объем виртуальной памяти, т.е. физическая память плюс *swap* — подкачка), *RSS* (объем физической памяти, занимаемой процессом), *TTY* (терминал), *STAT* (статус процесса), *START* (время начала), *TIME* (время работы процесса), *COMMAND* (команда, запустившая процесс).

Как можно видеть на рис. 1, значения столбца *COMMAND* у некоторых процессов даны в квадратных скобках. Такие процессы находятся в адресном пространстве ядра и предназначены для работы фоновых задач ядра. Управлять этими процессами нельзя.

Параметр *m* позволяет наблюдать потоки внутри каждого процесса. Как уже говорилось, каждому потоку выделяется собственная часть процессорного времени. Остальные ресурсы, включая память, принадлежат процессу: *ps auxm* (рис. 2).

С программой *ps* часто используется группа из двух UNIX-параметров — *-ef* (*ps -ef*): *-e* — показывать все процессы, *-f* — показывать в широком формате. Результат будет выведен уже в несколько другом наборе столбцов. Вы увидите еще и столбец с заголовком *PPID*. Это PID «родительского» процесса: когда один процесс вызывает другой, то вызвавший процесс становится родительским по отношению к вызванному. Вызванный процесс, в свою очередь, называется дочерним. По окончании своей работы дочерний процесс возвращает родительскому процессу код возврата. Завершение родительского процесса вызывает завершение всех его дочерних процессов. Вот почему при закрытии консоли прекращают работу все программы, в ней запущенные.

Отношение родительских и дочерних процессов можно нагляднее увидеть, если вызвать программу *ps* с параметром *fax*: *ps fax*. Здесь *f* означает отображение иерархии процессов (*forest*).

На рис. 3 показан вызванный нами процесс *ps fax*, который является дочерним по отношению к *Bash*. И это понятно: мы ведь

вызывали его с помощью командной оболочки *Bash*. Сама оболочка *Bash* была вызвана программой *mate-terminal*.

С помощью *ps* можно также отобразить информацию не обо всех, а только о некоторых процессах. UNIX-параметр *-C* позволяет сделать выборку по имени.

Откроем вторую консоль (чтобы на экране у нас их было две) и введем команду *ps -C bash*. В результате мы увидим, что процессов с именем *bash* всего два. Это понятно: ведь в каждой консоли работает отдельный экземпляр *Bash*.

PID	TTY	TIME	CMD
4837	pts/1	00:00:00	bash
4999	pts/0	00:00:00	bash

Теперь запустим во второй консоли какой-нибудь очень «тяжелый» процесс. Пусть это будет бесконечное копирование информации из устройства */dev/zero* в устройство */dev/null*. Устройство */dev/zero* — это «специальный файл», то есть файл связи с устройством, который на самом деле ни с каким реальным устройством не связан, а при чтении из него бесконечно выдаются нулевые биты. Устройство */dev/null* — это тоже «специальный файл», но при чтении из него он сразу выдает конец файла. При записи в него информация пропадает бесследно. Вот и давайте запустим процесс, который будет бесконечно копировать нули из одного устройства в другое. Воспользуемся для этого программой *dd*:

```
dd if=/dev/zero of=/dev/null
```

Она входит в пакет *coreutils* и предназначена для копирования информации с возможным ее преобразованием. Параметры *if*, *of* означают «input file» и «output file». После ввода этой команды приглашение *Bash* не появилось: командная оболочка ждет завершения работы программы. Но, как мы уже знаем, она его не дожидается: копирование нулей из */dev/zero* в */dev/null* — процесс бесконечный.

Вернемся в первую консоль и выведем информацию по обоим экземплярам *Bash* и процессу *dd*. Это легко сделать, потому что параметр *-C* может принимать не только одно значение, но и список значений. Программа *ps* позволяет также задать список столбцов, которые мы хотим вывести. Список столбцов задается с помощью UNIX-параметра *-o*, либо BSD-параметра *o*, либо GNU-параметра *--format*. Буквенные спецификаторы для каждого поля можно найти в справочной системе: *man ps*. Давайте отобразим следующие столбцы: *PID*, *TT* (терминал), *STAT* (статус), *CMD* (команда, вызвавшая процесс):

```
ps -C bash,dd -o pid,tt,stat,cmd
ps -C bash,dd o pid,tt,stat,cmd
ps -C bash,dd --format=pid,tt,stat,cmd.
```

Во всех этих случаях результат будет одинаковый:

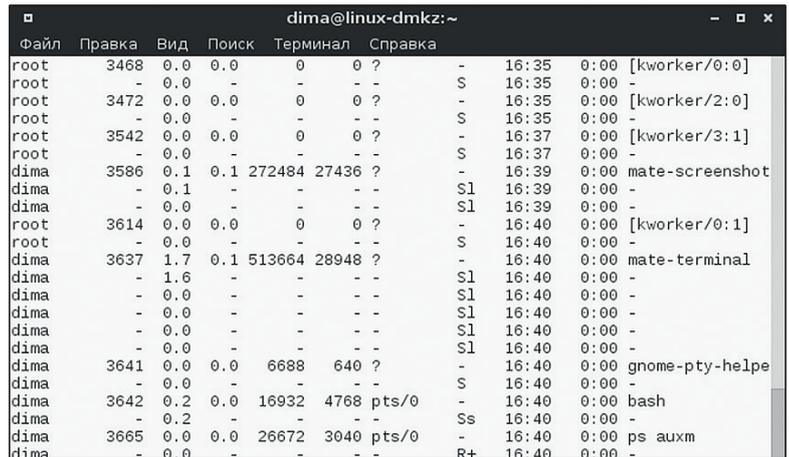
PID	TT	STAT	CMD
4837	pts/1	Ss	bash
4999	pts/0	Ss	bash
5021	pts/0	R+	dd id=/dev/zero of=/dev/null

Можно выборку делать не только по имени, но и по другим свойствам, например, по *PID*. Для этого используются параметры *-r* и *--pid*. Они равнозначны. Следующая команда выдаст абсолютно такой же результат, как приведенный выше:

```
ps r 4837,4999,5021 o pid,tt,stat,cmd.
```

Маленько заострим здесь внимание на параметре *STAT*. Это состояние (статус) процесса.

- Статусы бывают следующие.
- » D непрерываемый сон;
  - » R запущенный;
  - » S прерываемый сон или ожидание события;
  - » T остановлен сигналом;
  - » t остановлен отладчиком программы;
  - » W подкачка;
  - » X «мертвый»;



» Рис. 2. Результат команды *ps auxm*: можно видеть процессы и потоки.

» Z зомби: завершившийся, но еще не передавший родительскому процессу код завершения.

Если статус обозначается двумя символами, то добавляются следующие символы:

- » < высокий приоритет;
- » N низкий приоритет;
- » L имеет заблокированные страницы;
- » s лидер сессии;
- » l многопоточный;
- » + работает на переднем плане.

Итак, состояние процесса *dd*, запущенного нами во второй консоли, обозначено как *R+* (выполняется на переднем плане). Давайте изменим это. Вернемся в консоль, которую мы «подвесили» бесконечным процессом, и нажмем сочетание клавиш *Ctrl+Z*. Это сочетание «горячих клавиш» *Bash* означает команду остановить процесс, но не «убивать». Обратите внимание, что после нажатия *Ctrl+Z* в консоли появится слово *Stopped* с указанием команды, вызвавшей процесс, а затем — приглашение *Bash*. Когда процесс в остановленном состоянии, консоль доступна для работы.

После нажатия *Ctrl+Z* вернемся в предыдущую консоль и проверим статус этого процесса:

```
ps -C dd o pid,tt,stat,cmd
```

Результат покажет, что процесс «жив», а статус у него — остановлен (*T*).

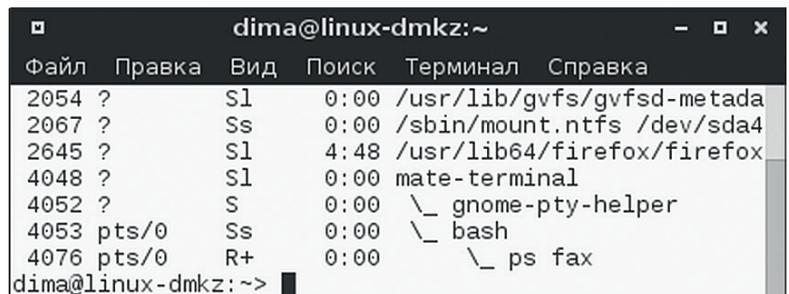
PID	TT	STAT	CMD
5021	pts/0	T	dd if=/dev/zero of=/dev/null

Теперь запустим его опять. Для этого вернемся в консоль с процессом *dd* и скомандуем *fg*. Консоль опять зависнет, а статус процесса опять станет *R+*.

*fg* — это внутренняя команда *Bash*, означающая запуск остановленного процесса на переднем плане. Но его можно запустить и на заднем плане (в фоне). Для этого служит команда *bg*.

Снова останавливаем процесс с помощью *Ctrl+Z* и комануем *bg*. На этот раз консоль не зависла: приглашение *Bash* появилось.

Снова вводим уже знакомую нам команду:



» Рис. 3. Иерархия процессов.

```

dima@linux-dmzkz:~
Файл Правка Вид Поиск Терминал Справка
top - 20:02:00 up 43 min, 3 users, load average: 0,99, 0,63, 0,35
Tasks: 210 total, 2 running, 208 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5,3 us, 19,8 sy, 0,0 ni, 74,5 id, 0,3 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem: 16109712 total, 1945992 used, 14163720 free, 888 buffers
KiB Swap: 2097148 total, 0 used, 2097148 free. 1138388 cached Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 2954 dima      20   0   5780   1308   1220 R 100,0 0,008  2:44.24 dd
1002 root        20   0 470620 78188 56876 S  0,667 0,485  0:36.49 Xorg
2326 dima      20   0 8195104 276588 176072 S  0,667 1,717  0:21.09 soffice.b+
341  root        20   0   0       0       0  S  0,333 0,000  0:00.32 btrfs-tra+
1572 dima      20   0 527152 27160 22336 S  0,333 0,169  0:00.15 clock-app+
1  root        20   0 336660 5844 3416 S  0,000 0,036  0:01.73 systemd
2  root        20   0   0       0       0  S  0,000 0,000  0:00.00 kthreadd
3  root        20   0   0       0       0  S  0,000 0,000  0:00.00 ksoftirqd+
5  root        0 -20   0       0       0  S  0,000 0,000  0:00.00 kworker/0+
7  root        20   0   0       0       0  S  0,000 0,000  0:00.52 rcu_preem+
8  root        20   0   0       0       0  S  0,000 0,000  0:00.00 rcu_sched
9  root        20   0   0       0       0  S  0,000 0,000  0:00.00 rcu_bh
10 root        20   0   0       0       0  S  0,000 0,000  0:00.34 rcuop/0
11 root        20   0   0       0       0  S  0,000 0,000  0:00.00 rcuos/0
12 root        20   0   0       0       0  S  0,000 0,000  0:00.00 rcuob/0
13 root        rt   0   0       0       0  S  0,000 0,000  0:00.00 migration+
14 root        rt   0   0       0       0  S  0,000 0,000  0:00.00 watchdog/0
    
```

Рис. 4. Программа top.

```
ps -C dd -o pid,tt,stat,cmd
```

Статус процесса будет R, но уже без плюса. Значит, наш процесс `dd` работает, но теперь на заднем плане.

Итак, процессы могут протекать на переднем и на заднем плане. В последнем случае еще говорят «фоновые процессы». Они бывают необходимы в тех случаях, когда требуется выполнять какую-нибудь долгосрочную задачу и чтобы можно было продолжать работать в консоли, не дожидаясь завершения этой задачи. Фоновые процессы не имеют возможности принимать команды от пользователя.

Можно ли вывести процесс с заднего плана на передний? Конечно, можно. Для этого опять перейдем в консоль, где мы запускали `dd`, и введем команду `fg`. Консоль нам полностью доступна, поэтому останавливать процесс не нужно.

Если сейчас посмотреть статус процесса, мы опять увидим: R+. А консоль с процессом `dd` опять зависла.

Теперь давайте полностью «убьем» этот процесс с помощью другого сочетания горячих клавиш: `Ctrl+C`. После этого, если мы попробуем вызвать информацию о нем с помощью `ps -C dd`, мы не получим никакого результата. «Убитый» процесс полностью удаляется из системы, и снова запустить его уже невозможно.

А можно ли изначально запустить процесс в фоне? Да, можно. Для этого команда должна завершиться знаком `&`:

```
dd if=/dev/zero of=/dev/null &
```

Вот теперь мы запустили эту задачу сразу на заднем плане. И поскольку у нас консоль доступна, мы можем запустить в фоне еще две «долгоиграющие» задачи:

```
cat > /dev/null < /dev/zero &
yes > /dev/null &
```

Рис. 5. Справка по программе top.

```

dima@linux-dmzkz:~
Файл Правка Вид Поиск Терминал Справка
Help for Interactive Commands - procsps-ng version 3.3.9
Window 1:Def: Cumulative mode Off. System: Delay 3,0 secs; Secure mode Off.

Z,B,E,e Global: 'Z' colors; 'B' bold; 'E'/'e' summary/task memory scale
l,t,m Toggle Summary: 'l' load avg; 't' task/cpu stats; 'm' memory info
0,1,2,3,I Toggle: '0' zeros; '1/2/3' cpus or numa node views; 'I' Irix mode
f,F,X Fields: 'f'/'F' add/remove/order/sort; 'X' increase fixed-width

L,&,<,> . Locate: 'L'/'&' find/again; Move sort column: '<'/'>' left/right
R,H,V,J . Toggle: 'R' Sort; 'H' Threads; 'V' Forest view; 'J' Num justify
c,i,S,j . Toggle: 'c' Cmd name/line; 'i' Idle; 'S' Time; 'j' Str justify
x,y . Toggle highlights: 'x' sort field; 'y' running tasks
z,b . Toggle: 'z' color/mono; 'b' bold/reverse (only if 'x' or 'y')
u,U,o,0 . Filter by: 'u'/'U' effective/any user; 'o'/'O' other criteria
n,#,^0 . Set: 'n'/'#' max tasks displayed; Show: Ctrl+^0 other filter(s)
C,... . Toggle scroll coordinates msg for: up,down,left,right,home,end

k,r Manipulate tasks: 'k' kill; 'r' renice
d or s Set update interval
W,Y Write configuration file 'W'; Inspect other output 'Y'
q Quit
(commands shown with '.' require a visible task display window)
Press 'h' or '?' for help with Windows,
Type 'q' or <Esc> to continue
    
```

Итак, три процесса — в фоне. Если мы сейчас введем команду `fg`, то перенесем на передний план последний из них. А можно ли проделать это не с последним процессом, а скажем, со вторым? Можно, если после `fg` указать его номер. Чтобы посмотреть все номера фоновых процессов, надо ввести команду `jobs`. Это внутренняя команда `Bash`, позволяющая посмотреть процессы, запущенные из текущей консоли.

В результате мы получим список из трех процессов. Как видим, последний из них помечен знаком «плюс», предпоследний — знаком «минус», остальные пометки не имеют.

```
[1] Running dd if=/dev/zero of=/dev/null &
[2]- Running cat > /dev/null < /dev/zero &
[3]+ Running yes > /dev/null &
```

Как видим, процесс `cat` имеет номер 2. А значит, мы знаем, как перевести его на передний план: `fg 2`.

Вызовем в соседней консоли программу `ps`, чтобы посмотреть состояние этих процессов:

```
ps -C dd,cat,yes -o pid,tt,stat,cmd
```

Теперь мы видим, что средний процесс работает на переднем плане, остальные — в фоне.

PID	TT	STAT	CMD
5676	pts/0	R	dd if=/dev/zero of=/dev/null
5700	pts/0	R+	cat
5703	pts/0	R	yes

## Программа top

Следующая программа этого пакета — `top`. Она работает в реальном времени в псевдографическом режиме и отображает процессы, отсортированные по загрузке процессора в порядке убывания. Естественно, в окне программы не всегда умещаются все процессы, но их список можно прокрутить вниз клавишей стрелки вниз, а обратно — клавишей стрелки вверх.

Чтобы проверить работу этой программы, давайте снова вызовем уже знакомую нам команду копирования нулей:

```
dd if=/dev/zero of=/dev/null
```

Теперь откроем вторую консоль и запустим там программу `top` (рис. 4). Мы видим, что процесс, созданный программой `dd`, выскочил в самый верх списка, поскольку он нагружает процессор больше всех.

Переходим в консоль с работающей программой `top` и нажимаем на клавишу `h`. Перед нами предстанет справка по программе (рис. 5). Давайте посмотрим по этой справке, как программой можно управлять.

В справке сказано, что клавишей `f` мы откроем страницу списка параметров, которые можно показывать или не показывать в программе. Выходим из справки с помощью клавиши `Esc` и снова оказываемся в главном окне программы. Теперь нажимаем `f`. И у нас открывается список параметров.

В этом списке жирным шрифтом выделены отображаемые параметры, а обычным шрифтом показаны не отображаемые. Клавишами стрелки вверх и стрелки вниз можно перемещаться по списку. Давайте переместимся к параметру `PPID` (идентификатор родительского процесса), и выберем его клавишей пробела. Он теперь должен выделиться жирным шрифтом. Далее с помощью клавиши стрелки вправо выделим его и клавишей стрелки вверх переместим его в самый верх. Затем снимем выделение клавишей стрелки влево.

Выйдем из режима выбора параметров с помощью клавиши `Esc`. Мы снова попадаем в главное окно программы. Крайний слева столбец у нас теперь — `PPID`. Возможно, при этом пропадет крайний справа столбец, так как он теперь может не уместиться в окне консоли. Но, чтобы его увидеть, достаточно прокрутить список горизонтально клавишами стрелки вправо и стрелки влево.

Теперь нажимаем клавишу `Z` (буква заглавная). И перед нами откроется окно выбора цвета. В этом окне есть список цветов

с их цифровыми кодами. Наждем на цифру 1 (код красного цвета), а затем Enter. Мы снова попадаем в главное окно программы, и все столбцы теперь окрашены в красный цвет (рис. 6).

Посмотрите, какие еще возможности описаны в справке, открываемой клавишей h. Так, например, клавишей g (genice) можно изменить свойство nicepess процесса. Это свойство отображается в столбце NI. У большинства процессов это свойство равно 0. Чем выше значение этого свойства, тем чаще данный процесс будет уступать процессорное время другим процессам. То есть понимать его можно как «доброта». Значения этого свойства могут быть от -20 (наибольший приоритет) до +19 (наименьший приоритет). Установить nicepess меньше текущего можно, только имея права суперпользователя.

У нас всё еще работает процесс dd. Давайте сделаем его чуть-чуть «добрее». Нажимаем клавишу g. В верхней части окна появляется приглашение ввести PID нужного процесса. Вводим PID процесса dd, нажимаем Enter. После этого появится приглашение ввести новое значение nicepess. Вводим 10, нажимаем Enter. Теперь в столбце NI напротив процесса dd должно появиться значение 10.

Клавиша k позволяет убить, остановить, перезапустить процесс — в общем, послать ему сигнал. Нажимаем k. Появится приглашение ввести PID процесса. Вводим PID, нажимаем Enter. Затем появится приглашение ввести номер сигнала. Если мы не будем вводить номер сигнала, а сразу нажмем второй раз Enter, программа pošлет процессу сигнал № 15 (TERM). Нажав Enter, мы увидим, что процесс dd исчез из списка процессов программы top. А в соседней консоли появилось слово «Завершено» и приглашение командной строки. Процесс убит.

## Программа pidof

Переходим к следующей программе пакета. Программа pidof, как говорит само ее название, предназначена для определения PID какого-либо процесса. Если у нас сейчас всё еще запущена программа top, то в соседней консоли мы можем посмотреть ее PID: pidof top.

Программа pidof может принимать несколько параметров. Они вводятся друг за другом, разделяясь пробелами. Давайте заодно посмотрим PID запущенных экземпляров Bash. В результате команды pidof top bash у нас будет выведен список PID процессов с указанными именами.

## Программа pwdx

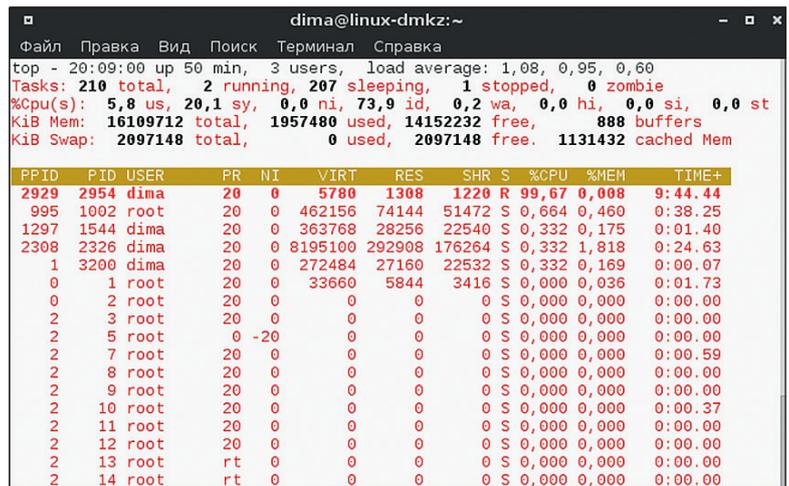
Эта программа определяет рабочий каталог процесса. Проверяем ее работу. Запускаем фоновый процесс:

```
yes > /dev/null &
    Создаем новый каталог и переходим в него: mkdir inner; cd inner.
    Запускаем второй фоновый процесс:
dd if=/dev/zero of=/dev/null &
    Выясняем PID наших фоновых процессов — pidof yes dd — и определяем их рабочие каталоги, указав полученные PID:
pwdx 3960 3976.
    Результат должен получиться такой:
3960: /home/dima
3976: /home/dima/inner
```

## Программа w

Программа w определяет пользователей, находящихся в системе, и их процессы. В заголовке отображается следующая информация: текущее время; время работы системы; сколько пользователей находятся в системе; средняя загрузка системы за последние 1, 5 и 15 минут. Под средней загрузкой системы понимается среднее количество процессов в статусе R (работа) и S (ожидание события).

Для каждого пользователя сообщается следующее: имя, терминал, удаленный хост (если этот пользователь зашел удаленно),



► Рис. 6. Мы добавили столбец PPID и окрасили программу в красный цвет.

время пребывания в системе; время неактивности; время работы всех процессов, связанных с данным терминалом; время работы указанного процесса; название указанного процесса.

Из рис. 7 видно, что пользователь зашел в систему несколько раз, но через разные терминалы. Виртуальных терминалов в системе несколько. Но только в одном из них, в нашей системе, запущена графическая оболочка. В остальных — вся работа идет в режиме командной строки. (Впрочем, в других системах она может не быть запущена вообще нигде.) Переключаться между терминалами можно с помощью сочетания клавиш Alt+F1, Alt+F2, Alt+F3 и т. д., в зависимости от номера терминала, в который мы хотим переключиться. Но если мы находимся в терминале с графической оболочкой, то она может «перехватить» сочетания с клавишей Alt, назначив им совсем другие действия. Поэтому из графического терминала в неграфический надо переключаться с помощью сочетания трех клавиш: Ctrl+Alt+F1, Ctrl+Alt+F2 и т. д.

В графической оболочке можно запустить программную консоль, и ей будет сопоставлено другое виртуальное устройство: псевдотерминал.

Во многом эта программа сходна с программой who из пакета coreutils. Разница — в частных деталях.

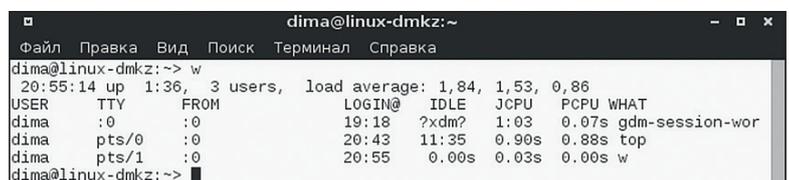
## Программа uptime

Программа uptime абсолютно сходна с аналогичной программой из пакета coreutils. Поэтому в проекте Linux From Scratch (LFS, «Линукс с нуля») автор рекомендует собирать пакет coreutils с опцией --enable-no-install-program=kill,uptime, означающей, что в сборку не надо включать программы kill и uptime, потому что uptime будет установлен с пакетом procps-ng, а kill — с пакетом util-linux.

Задача программы — вывести текущее время, время работы системы, количество пользователей в системе, и среднее количество процессов в статусе R и S за последние 1, 5 и 15 минут.

```
uptime
21:04 в работе 1:46, 3 пользователя, средняя загрузка:
0.06,0.27,0.49
```

Вот мы и узнали немало интересного о процессах GNU/Linux. На следующем уроке продолжим знакомство с пакетом procps-ng. **LXF**



► Рис. 7. Вывод информации о пользователях по команде w.

**Скорая помощь**

Для запуска процесса в фоновом режиме необходимо ввести команду со знаком &. Просмотреть все процессы, запущенные в фоновом режиме из данной консоли, можно командой jobs.

# VFS: Файловые системы Linux

Джонни Бидвелл продолжает учебу у Linux Foundation. Тема этого урока — многочисленные файловые системы Linux и их объединение в одну VFS.



**Наш эксперт**

С чувством времени у **Джонни Бидвелла** так себе. Ну и ладно, на это есть *cron*. На беду, Джонни не разбирается в его синтаксисе.

Linux поддерживает великое множество файловых систем (ФС). Возможно это становится благодаря уровню абстракции Virtual File System (VFS), находящемуся между ПО и дисковой файловой системой.

Для корпоративных дистрибутивов стали стандартом расширенные журналируемые файловые системы (ФС). Приложения читают и записывают файлы, а не их физические байты на машине, где они хранятся. Файлы и их имена — это абстракция, маскирующая физический уровень ввода-вывода. Прямая запись на диск — помимо уровня файловой системы — опасна и выполняется только с помощью низкоуровневых программ операционной системы, и ни в коем случае не пользователями.

Локальные файловые системы находятся в одном из разделов диска, который может быть физическим или логическим, управляемым диспетчером логических томов (LVM). Также ФС могут быть сетевыми, когда их физическое воплощение скрыто от локальной системы и распределено по всей сети.

Во всех системах Linux используется топология инверсного дерева, отделяющая корневую директорию (/). Всё дерево может храниться в одной локальной файловой системе и в одном разделе, но как правило, создается несколько разделов (или сетевых файловых систем), соединенных вместе в точках монтирования. Сюда также могут включаться съемные носители, такие как USB-накопители и оптические диски.

Кроме того, некоторые виртуальные файловые системы монтируются внутри дерева. К ним относятся */proc*, */sys* и */dev* и, возможно, */tmp* и */run*. По сути, каждый из элементов, установленных внутри дерева, может иметь собственную файловую систему. Но для приложений и операционной системы всё это существует в виде единой древовидной структуры.

В Linux реализована виртуальная файловая система (VFS). Когда приложению требуется доступ к файлу, оно взаимодействует

с уровнем абстракции VFS, который затем преобразует все системные действия ввода/вывода (чтение, запись и т.д.) в конкретный код, относящийся к конкретной файловой системе. Таким образом, приложения не касаются ни самой по себе файловой системы, ни физических носителей и аппаратных средств, на которых она находится. Кроме того, сетевые файловые системы могут работать прозрачно. Благодаря этому, Linux справляется с гораздо большим числом файловых систем, чем любая другая операционная система.

Большинство файловых систем имеют полный доступ на чтение и запись; у некоторых есть доступ только на чтение и, возможно, экспериментальный доступ на запись. Для некоторых типов файловых систем, особенно тех, что основаны не на Unix, может потребоваться больше манипуляций, чтобы быть представленными в VFS. Например, в *vfat* нет явных разрешений на чтение/запись/выполнение (*rwX*) для полей владелец/группа/все пользователи. Потому VFS приходится домысливать, как должны быть указаны разные разрешения для трех типов пользователей, а зависеть это может и от способа монтирования. Существуют файловые системы вне ядра, такие как *ntfs-3g* с поддержкой записи и чтения ([www.tuxera.com/community/ntfs-3g-download](http://www.tuxera.com/community/ntfs-3g-download)); они надежны, но их производительность слабее, чем у обычных.

## ФС в ассортименте

Наиболее часто используемые файловые системы — *ext4*, *xf*s, *btrfs*, *squashfs*, *nfs* и *vfat*. В таблице ниже приведен неполный список поддерживаемых файловых систем:

Название	Описание
<i>ext4</i> , <i>ext3</i> , <i>ext2</i>	«Родные» файловые системы Linux
<i>proc</i>	Используется для <i>/proc</i>
<i>vfat</i>	Windows VFAT (включая FAT32, FAT и т.д.)
<i>ntfs</i>	Windows NT и новее (только для чтения)
<i>udf</i>	CD R/W, DVD
<i>hfs+</i>	Apple Extended HFS
<i>jffs</i> , <i>jffs2</i>	Journalling Flash Filesystem
<i>iso9660</i>	CD-ROM, включая расширения Joliet
<i>tmpfs</i>	Заменяемый RAM-диск
<i>gfs2</i>	Файловая система на базе кластеров, от Red Hat
<i>nfs</i>	Network Filesystem (с версии 4)
<i>smb</i>	Сетевая, Samba
<i>ncp</i>	Novell Netware FS с использованием протокола NCP
<i>coda</i>	Экспериментальные распределенные файловые системы
<i>afs</i>	Распределенная ФС Andrew, от Университета Карнеги-Меллона
<i>ocfs2</i>	Кластерная файловая система на основе экстенстов, от Oracle

```

Activities
nodev hugetblfs
nodev devpts
nodev autofs
nodev pstore
nodev elfvarfs
nodev iqueue
nodev ext3
nodev ext2
nodev ext4
nodev btrfs
nodev fuseblk
nodev fuse
nodev fusectl
l.jonnib@acerotspades ~$ ls -lh /sbin/mkfsx
-rwxr-xr-x 1 root root 11K Mar 26 22:54 /sbin/mkfs
-rwxr-xr-x 1 root root 27K Mar 26 22:54 /sbin/mkfs.bfs
-rwxr-xr-x 1 root root 35K Mar 26 22:54 /sbin/mkfs.cramfs
lrwxrwxrwx 1 root root 9 Jun 9 14:34 /sbin/mkfs.extfat -> mkextfatfs
-rwxr-xr-x 4 root root 121K Feb 1 09:01 /sbin/mkfs.ext2
-rwxr-xr-x 4 root root 121K Feb 1 09:01 /sbin/mkfs.ext3
-rwxr-xr-x 4 root root 121K Feb 1 09:01 /sbin/mkfs.ext4
-rwxr-xr-x 1 root root 31K Mar 9 07:46 /sbin/mkfs.fat
-rwxr-xr-x 2 root root 55K May 16 2013 /sbin/mkfs.jfs
-rwxr-xr-x 1 root root 88K Mar 26 22:54 /sbin/mkfs.minix
lrwxrwxrwx 1 root root 8 Mar 9 07:46 /sbin/mkfs.msdos -> mkfs.fat
lrwxrwxrwx 1 root root 18 Mar 29 19:34 /sbin/mkfs.ntfs -> /usr/bin/mkntfs
lrwxrwxrwx 1 root root 18 Jun 9 2015 /sbin/mkfs.reiserfs -> mkreiserfs
lrwxrwxrwx 1 root root 8 Mar 9 07:46 /sbin/mkfs.vfat -> mkfs.fat
-rwxr-xr-x 1 root root 441K May 22 14:59 /sbin/mkfs.xfs
l.jonnib@acerotspades ~$
    
```

➤ Файловых систем так много, а жизнь коротка. **ExFAT** универсальна, как и **FAT**, но не ограничена досадными 2 Гб.



Подпишитесь сейчас и сэкономьте 10%

## Учитесь с Linux Foundation

Подпишитесь на любой учебный курс Linux Foundation сегодня и получите скидку 10%\* от обычной цены! Эксперты из Linux Foundation создали курсы, охватывающие весь спектр Linux: от классических курсов Sysadmin до новых технологий, таких как OpenStack и Open Daylight.

Если вам понравился этот небольшой образчик и вы хотите узнать больше — мы взяли эту часть из очень популярного курса *LFS201 Essentials*

*of System Administration*, который дает дополнительную квалификацию Linux Foundation Certified Sysadmin. Курс полностью интерактивен, самостоятелен и имеет более 80 экспертных лабораторных работ с решениями для вас для тренировки ваших новых навыков.

Чтобы получить 10-процентную скидку на ЛЮБОЙ курс, выполните следующие действия: зайдите на <https://training.linuxfoundation.org/linux-courses>

и просмотрите весь спектр имеющихся курсов.

Выберите тот, который вас интересует, например, популярный *LFS201 Essentials of System Administration*.

Наконец, нажмите кнопку Register и создайте учетную запись, заодно введя код купона\* **TECH10**, чтобы получить 10-процентную скидку. И вы уже студент!

\* Действует до 30.12.2017

В ряде новых высокопроизводительных файловых систем есть функция полного журналирования. Журналируемые файловые системы восстанавливаются после сбоев системы либо непредвиденных отключений с минимальными потерями или вовсе без них, и делают это очень быстро. Это требует некоторых дополнительных операций, но расширение возможностей позволяет минимизировать лишние затраты. В журналируемой ФС операции группируются в транзакции. Транзакция должна быть выполнена без ошибок, атомарно — в противном случае ФС не будет изменена. Сохраняется журнал транзакций. При возникновении ошибки обычно достаточно проверить только последнюю транзакцию.

Файловая система ext3 — это расширение исходной файловой системы Linux ext2. Сейчас на смену обеим пришла ext4, функциональность которой включает экстенты, 48-битные номера блоков и файлы размером до 16 ТБ. В большинстве дистрибутивов Linux ext4 используется в качестве файловой системы по умолчанию уже несколько лет. Среди прочих отраслевых файловых систем — IBM JFS и SGFS XFS (последняя является файловой системой по умолчанию в RHEL 7). Новейшая из журналируемых файловых систем, btrfs, всё еще находится в разработке. Это файловая система по умолчанию в SUSE Linux Enterprise Server 12.

Список типов файловых систем, зарегистрированных и распознаваемых текущей версией ядра Linux, можно посмотреть, выполнив `$ cat /proc/filesystems`.

Обратите внимание, что дополнительные файловые системы могут загружать свой код только тогда, когда система пытается получить доступ к разделу, который их использует. В Linux специальные файловые системы широко применяются для решения определенных задач. Это особенно эффективно для работы с различными структурами данных ядра, настройки его поведения или выполнения отдельных функций. Примеры можно увидеть в приведенной выше таблице. Заметим, что некоторые из этих специальных файловых систем не имеют точки монтирования; это означает, что пользовательские приложения с ними не взаимодействуют, но их использует ядро, посредством слоев VFS и кода.

## Свойства файловой системы

В число важных функций ФС входят индексные дескрипторы [inode], файлы каталогов, а также жесткие и мягкие ссылки. Традиционные разрешения файлов Unix расширяются за счет дополнительных атрибутов. В разных ФС есть свои утилиты для выполнения таких задач, как создание и форматирование, проверка и исправление ошибок, а также установка и размонтирование в ходе загрузке или после нее. Индексный дескриптор — это структура данных на диске, которая описывает и сохраняет атрибуты файлов, включая местоположение. Каждому файлу присваивается собственный индекс, содержащий информацию о правах доступа, пользователь/группе, которые им владеют, его размере и временных метках (время последнего доступа, время модификации и время изменения). Имени файла там нет, оно хранится в файле каталога. Также в индексном дескрипторе обычно хранятся все

операции ввода-вывода в отношении файла, поскольку эта информация должна обновляться. Файл каталога — это особый тип файла, связывающий имена файлов и их индексные дескрипторы.

Существует два способа связать имя файла с индексом: жесткие ссылки, указывающие на сам индексный дескриптор, и мягкие (символические) ссылки, которые указывают на имя файла, у которого есть связанный индексный дескриптор. Каждая привязка содержимого файла каталога и индексного дескриптора называется ссылкой. Создать дополнительные ссылки можно посредством команды `ln`.

Поскольку в каталоге может быть две или более записей, указывающих на один и тот же индекс (жесткие ссылки), у файла может быть несколько имен, каждому из которых отводится место в структуре каталогов. Однако индексный дескриптор может быть только один — независимо от того, какое имя используется. Когда процесс ссылается на имя пути, ядро ищет в каталогах соответствующий номер индекса. После того, как имя преобразовано в номер индексного дескриптора, последний загружается в память и используется в последующих запросах.

Расширенные атрибуты связывают с файлами метаданные, не интерпретируемые напрямую файловой системой. Существует четыре пространства имен: пользовательское, доверенное, безопасное и системное. Пространство имен системы используется для списков управления доступом (ACL), а пространство имен безопасности используется в SELinux. Флаговые значения хранятся в индексном дескрипторе и могут быть изменены и заданы только суперпользователем-root.

У каждого типа файловой системы есть утилита для форматирования (создания) файловой системы внутри раздела. Общее имя для этих утилит — `mkfs`. Однако это просто интерфейс для различных программ, своих для каждой файловой системы, которые можно просмотреть с помощью `ls -lh/sbin/mkfs*`. У любой файловой системы есть собственные параметры, задаваемые при форматировании. Например, при создании файловой системы ext4 следует помнить о настройках журналирования. К ним относятся задание размера файла журнала и возможность использования внешнего файла. Детали можно посмотреть на map-страницах для программ `mkfs*`. **LXF**

## Скорая помощь

Все устанавливаемые флаги атрибутов можно посмотреть по map chattr. Формат их установки — `chattr [+|-mode] filename`. В пространстве имен пользователей можно установить `i` (неизменяемый), `a` (только для добавления), `d` (без дампа) и `A` (без обновления atime). Флаги можно просмотреть с помощью `lsattr filename`.

Table 11.2: Special Filesystems

Filesystem	Mount Point	Purpose
roots	None	During kernel load, provides an empty root directory.
hugetlbfs	Anywhere	Provides extended page access (2 or 4 MB on X86).
bdev	None	Used for block devices.
proc	/proc	Pseudo filesystem access to many kernel structures and subsystems.
sockfs	None	Used by BSD Sockets.
tmpfs	Anywhere	RAM disk with swapping, re-sizing.
shm	None	Used by System V IPC Shared Memory.
pipefs	None	Used for pipes.
binfmt_misc	Anywhere	Used by various executable formats.
devpts	/dev/pts	Used by Unix98 pseudo-terminals.
usbfs	/proc/bus/usb	Used by USB sub-system for dynamical devices.
sysfs	/sys (or elsewhere)	Used as a device tree.
debugfs	/sys/kernel/debug (or elsewhere)	Used for simple debugging file access.

► Специальные файловые системы в Linux. Их море, гораздо больше, чем в этой полезной таблице.

# Rescatux: Спаси и сохрани

Маянк Шарма запускает Rescatux — универсальный спасательный плот для драгоценных данных на вашем компьютере, будь тот с Linux или Windows.



## Наш эксперт

**Маянк Шарма** является техническим писателем и бывшим редактором [Linux.com](http://Linux.com), но теперь коротает время, весь день сидя в трусах и играя в игры на Linux.

## Скорая помощь

Не удаляйте пароль для учетной записи с зашифрованными данными, используя пароль для входа в систему: могут случиться *Bad Things™* [Всеякие Пакоستي].

**М**ногое может испортить ваш компьютер. Неосторожное нажатие клавиши может удалить файл, переписать загрузчик, испортить пароль для входа или вызвать другие ужасные проблемы. Оказавшись в подобной ситуации, подавите панику и возьмите дистрибутив Rescatux ([www.supergrubdisk.org/rescatux](http://www.supergrubdisk.org/rescatux)). Этот Live CD предлагает богатую коллекцию инструментов, пригодных для решения широкого круга проблем в вашей установке Linux. Кроме того, Rescatux также пригодится, если у вас двойная загрузка, и может также исправить некоторые типовые проблемы с разделами Windows.

Дистрибутив собрал все важные и полезные инструменты для устранения неполадок с загрузкой Linux и Windows, включая *testdisk*, *photorec* и *GParted*. Rescatux можно использовать для восстановления MBR и загрузчиков, исправления ошибок файловой системы и таблиц разделов, а также сброса паролей в установленных Linux и Windows. Он содержит еще и инструменты для спасения данных и восстановления файлов и даже способен надежно стереть установки и Windows, и Linux.

Существует несколько спасательных дистрибутивов, но Rescatux превосходит их всех своей прямолинейностью. В отличие от других решений, предлагающих только набор инструментов для ремонта неисправного компьютера, Rescatux использует свое собственное приложение с кнопками, распределенными по категориям, которое поможет вам справиться с конкретными вашими проблемами.

Дистрибутив предназначен для использования в качестве «живой» среды, поэтому перенесите ISO на оптический диск или съемный USB-накопитель. Вставьте или подключите загрузочный носитель Rescatux на поврежденном компьютере и загрузитесь с него. Rescatux откроет минимальный графический рабочий стол *LXDE*. И сразу автоматически запустится собственное вспомогательное приложение под названием *Rescapp*.



➤ Rescatux поможет вам оправиться от беды. Он также содержит утилиты *GPGV* и *shred*, чтобы защитить вашу систему и предотвратить нечаянные утечки конфиденциальной информации.

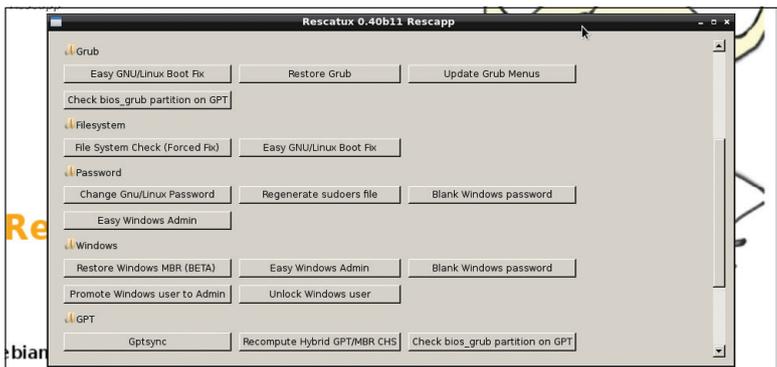
Интерфейс приложения улучшался от выпуска к выпуску, и его последняя версия имеет несколько кнопок, разделенных на различные категории, такие как Grub, Filesystem, Boot и Password. Кнопки внутри каждой категории имеют описательные метки, помогающие определить их функции. По нажатию кнопки появляется соответствующая документация, где подробно объясняется, какие шаги предпримет Rescatux и какую информацию он ожидает от пользователя. После прокрутки иллюстрированной документации, зная, чего ожидать, нажмите кнопку с надписью Run! для запуска соответствующей утилиты.

## Сперва fsck

Хотя файловые системы последнее десятилетие развивались довольно прытко, иногда, чтобы испортить жесткий диск, достаточно глючной программы, не оставляющей другой возможности, кроме как принудительно перезагрузить компьютер.

При перезагрузке, когда ваш дистрибутив Linux обнаруживает некорректное завершение работы, он автоматически запускает утилиту проверки файловой системы *fsck* для проверки целостности файловой системы. Во многих ситуациях это должно помочь. Но иногда, в зависимости от таких факторов, как возраст диска, файловая система и прерванное задание, автоматическая проверка не срабатывает.

В таком случае ваш дистрибутив попросит вас запустить инструмент *fsck* вручную. Хотя вы можете запустить *fsck* из режима обслуживания с монтированием файловой системы только для чтения, лучше запустить *fsck* с Live CD без монтирования раздела. Для этого загрузите Rescatux и выберите функцию File System Check (Forced Fix). Она проверит ваш компьютер и перечислит все разделы. Выберите тот, который выдавал ошибку, и Rescatux просканирует и исправит все несоответствия.



➤ В *Rescapp* есть несколько меню, вызывающих интерфейсы утилит командной строки, в которых используется мастер для перехода на различные этапы процесса.

## Решение для ремонта одним касанием

Многие проблемы с загрузчиком *Grub2* можно решить одним нажатием кнопки благодаря *Boot-Repair*.

Программа имеет интуитивно понятный интерфейс пользователя, умеет сканировать и понимать различные разметки дисков и схемы разбиения на разделы, а также может определить и правильно идентифицировать существующие на них установки операционных систем. Утилита способна работать как на традиционных компьютерах

с MBR, так и на современных компьютерах с UEFI и разметкой GPT.

*Boot-Repair* доступен в категории Expert Tools утилиты *Rescapp*. После запуска приложение сканирует жесткий диск, прежде чем показать свой простой интерфейс. Большинству пользователей достаточно следовать рекомендациям инструмента и просто нажать кнопку Recommended repair, что должно исправить большинство испорченных загрузчиков.

Восстановив загрузчик, инструмент выдаст ссылку на URL, который следует принять к сведению. URL содержит подробный отчет о дисках, разделах и содержимом важных файлов *Grub 2*, включая */etc/default/grub* и *boot/grub/grub.cfg*. Если инструмент не сумел устранить проблему с загрузчиком, вы можете поделиться URL на форумах вашего дистрибутива. Это позволит другим людям разобраться в разметке вашего диска и дать вам полезные советы.

## Загрузчик

Одной из наиболее распространенных проблем, преследующих пользователей Linux, является испорченный загрузчик. Чтобы превратить компьютер в кирпич, много трудиться не надо. Основная загрузочная запись (MBR) расположена в специальной области в начале каждого жесткого диска. Она помогает отслеживать физическое расположение всех разделов, а также содержит загрузчик. Всё, что требуется для стирания MBR — это нажать не ту клавишу в *fdisk* или *gparted*.

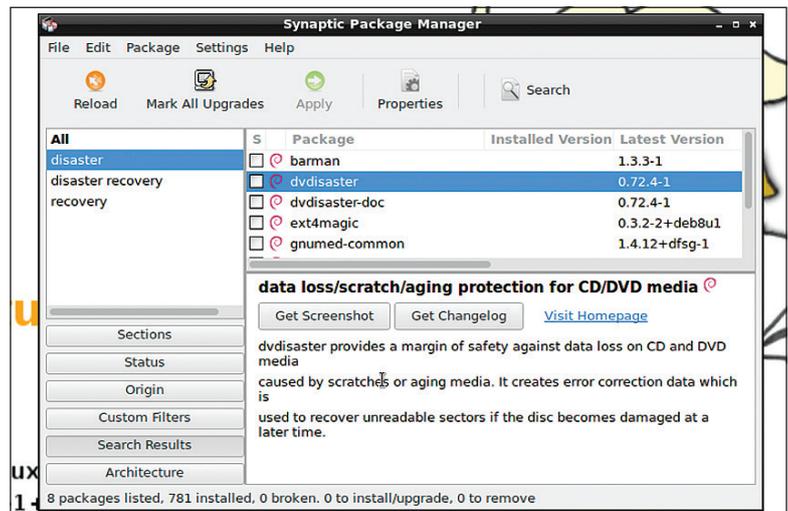
*Rescatux* включает несколько вариантов восстановления загрузчика *Grub* и исправления MBR. Функция *Restore Grub* сначала сканирует все разделы на компьютере и считывает их идентификационную информацию из файла */etc/issue*. Затем она отобразит их список и попросит выбрать ваш основной дистрибутив Linux. Потом она проверяет все диски, подключенные к компьютеру, и просит выбрать тот, где вы хотите установить *Grub*. Если у вас несколько дисков, меню предложит вам изменить их порядок в соответствии с порядком загрузки. По предоставлению всей этой информации запустится команда *grub-install* из выбранного дистрибутива Linux, она создаст новый загрузчик и поместит его в MBR. Если у вас дистрибутив на базе Debian, такой как Ubuntu, вы можете использовать функцию, обозначенную Update GRUB Menus. Она проведет вас через те же шаги, что и функция *Restore Grub*, но оптимизирована для добавления в загрузчик *Grub* дистрибутивов на базе Debian.

Новые версии дистрибутива также включают функцию *Ease GNU/Linux Boot fix*, которая предлагает комбинацию из трех вариантов. Она начинается принудительной проверкой файловой системы перед запуском функции обновления *Grub* и заканчивается функцией восстановления *Grub*. На первый взгляд вам по-прежнему будет предложено только выбрать основной дистрибутив Linux и диск, на котором должен храниться загрузчик *Grub*. За кулисами этот новый вариант использует информацию для трех упомянутых задач.

## Сезам, откройся

Это случалось со всеми. Удумывание невразумительного пароля не принесет вам пользы, если вы не сможете его запомнить. Вместо бесконечных попыток попасть в яблочко путем перестановок и комбинаций можно обратиться к *Rescatux* и без особых усилий задать себе новый пароль. Дистрибутив предлагает опции восстановления пароля для Linux и Windows.

Если вы забыли пароль для своей установки Windows, запустите *Rescatux* и выберите функцию *Blank Windows Password* в утилите *Rescapp*. Затем дистрибутив ищет на компьютере разделы, содержащие файл Security Account Manager (SAM). Обычно он показывает только один раздел, но если у вас несколько вариантов



Windows, мастер отобразит несколько разделов. Выберите раздел, содержащий пользователя, пароль которого вы хотите восстановить. Затем *Rescatux* создаст резервные копии файлов SAM и отобразит список пользователей, найденных в выбранном разделе. Выберите пользователя, пароль которого вы намереваетесь сбросить, и *Rescatux* уничтожит его пароль. Затем можно перезагрузить компьютер, перезагрузиться в Windows и войти как пользователь, которого вы только что «зачистили»; и Windows позволит вам это без запроса пароля.

Аналогично можно использовать функцию *Promote Windows user to Admin*, чтобы сделать именно это. Эта опция также сканирует и перечисляет все разделы Windows, в которых есть файл SAM. Выберите тот, который вас интересует, чтобы просмотреть список пользователей. Выберите пользователя из этого списка, и *Rescatux* настроит Windows так, чтобы предоставить ему те же привилегии, что и администратору. Этот механизм работает для всех версий Windows, включая Windows 10, если они защищены паролем. Однако это не сработает, если вы выбрали другой механизм блокировки учетной записи, такой как PIN-код.

Новая версия *Rescatux* включает опцию *Easy Windows Admin*. Эта функция предоставляет возможность вернуть себе контроль над установкой Windows, объединив несколько функций, чтобы сначала очистить пароль пользователя, а затем передать его администратору Windows.

Вы также можете использовать *Rescatux* для изменения паролей в установке Linux и восстановления поврежденного файла *sudoers*. Выберите функцию *Change GNU/Linux Password*, чтобы позволить *Rescatux* найти на компьютере установки Linux. Затем

► **Rescatux основан на Debian и включает диспетчер пакетов Synaptic, пригодный для установки дополнительных средств аварийного восстановления.**

**Скорая помощь**

Используйте функцию расширенного поиска *TestDisks* для сканирования каждого цилиндра и суперблоков, чтобы найти потерянные разделы, если опция *Quick Search* не может их откопать.

» Подпишитесь на печатную или электронную версии на [www.linuxformat.ru/subscribe!](http://www.linuxformat.ru/subscribe!)

выберите интересующий вас раздел Linux, чтобы просмотреть список пользователей в этой конкретной установке. Пользователь `root` находится вверху, а обычные учетные записи пользователей, которые вы создали во время установки или вручную после этого, перечислены внизу. В отличие от сброса пароля Windows, когда вы выбираете пользователя Linux, Rescatux дает вам возможность задать новый пароль.

Когда вы выбираете параметр `Regenerate sudoers file`, появляется похожий мастер. Он также ищет установку Linux, а затем отображает список пользователей в выбранном дистрибутиве. Однако вместо генерации пароля эта опция добавит выбор пользователя в список `/etc/sudoers`, что позволит ему запускать приложения с правами суперпользователя.

## Восстановление разделов

Иногда проблемы с диском гораздо серьезней, чем поврежденные MBR и загрузчик. Скачок напряжения, неисправный жесткий диск или неуклюжий оператор могут легко стереть таблицу разделов. *TestDisk* — лучший инструмент, который исправит таблицы разделов и сделает диски снова загружаемыми. Инструмент можно запустить из раздела `Expert` утилиты *Rescapp*.

При запуске *TestDisk* сначала попросит создать журнал (который пригодится для последующего анализа при неудаче восстановления), а затем отобразит список всех дисков, подключенных к компьютеру. После выбора диска, на котором вы потеряли раздел, он попросит вас выбрать тип таблицы разделов, такой как Intel, Mac, Sun и т. д.

Затем вам покажут различные варианты восстановления. Выберите функцию по умолчанию `Analyse`, которая считывает структуру разделов и ищет потерянные разделы. Отобразится текущая структура разделов. Теперь выберите функцию `Quick Search`, чтобы велеть *TestDisk* искать удаленные разделы. После этого появится список потерянных разделов. В зависимости от возраста вашего

диска *TestDisk* может отображать несколько разделов. Чтобы определить, какой именно раздел необходимо восстановить, найдите метку раздела в конце каждой записи в [квадратных скобках]. Если это вам не поможет, нажмите `P` на выбранном разделе, чтобы просмотреть список файлов, найденных *TestDisk* на этом разделе. Повторяйте это со всеми разделами, пока не найдете нужный раздел.

Найдя свой раздел, лучше всего скопировать данные — на случай, если *TestDisk* не сможет его восстановить. Для этого нажмите `P`, а затем с помощью клавиши `A` выберите все файлы. Потом нажмите `C`, чтобы скопировать файлы — вам предложат определить место для сохранения файлов. После этого нажмите `Q`, чтобы вернуться к списку восстановленных разделов, и нажмите `Enter` для перехода к следующему шагу восстановления выбранного раздела. *TestDisk* снова отобразит структуру разделов, на сей раз с учетом потерянного раздела. Теперь выберите `Write`, чтобы сохранить таблицу разделов на диске и выйти из программы. Если всё пройдет хорошо, то после перезагрузки компьютера ваш раздел будет восстановлен там, где он должен быть.

## Восстановление файлов

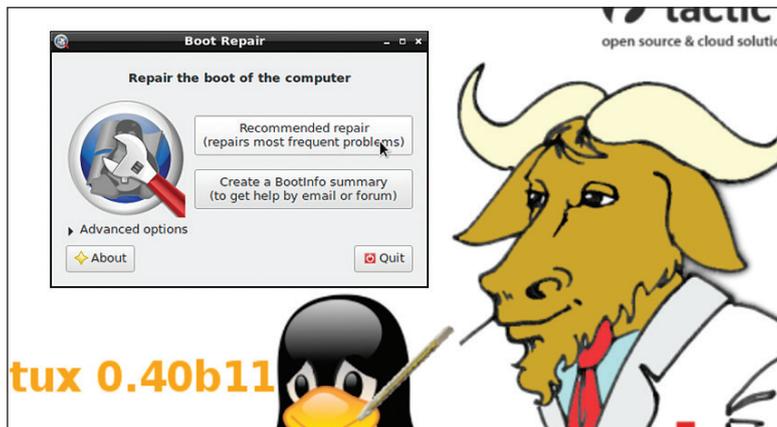
Кроме восстановления загрузки компьютеров, Rescatux помогает восстановить случайно удаленные файлы, поскольку потерю данных вы не всегда можете свалить на свои оборудования. Категория `Expert Tools` также включает утилиту *Photorec*, способную восстанавливать файлы даже при отсутствии обычных метаданных, потому что вместо того, чтобы полагаться на файловую систему, она тщательно сканирует весь жесткий диск.

Когда вы удаляете файл, на деле он не стирается полностью. Точнее, файловая система просто помечает его как удаленный и делает пространство, занимаемое файлом, доступным для других файлов. Это означает, что пока другая программа не воспользуется этим недавно освобожденным пространством, исходный файл всё еще существует и может быть восстановлен с помощью инструмента восстановления файлов, такого как *Photorec*. По этой причине, как только вы сообразите, что случайно удалили файлы, очень важно немедленно прекратить работу на компьютере, чтобы свести к минимуму взаимодействие с жестким диском.

Инструмент работает на всех дисках, включая жесткие диски и съемные носители, такие как USB-диски. Помимо чтения незагружаемых дисков, *Photorec* также восстановит файлы из переформатированных и переустановленных разделов. *Photorec* может находить наиболее распространенные форматы изображений и может дополнительно выделять файлы в различных форматах, включая ODF, PDF, 7zip, ZIP, TAR, RPM, Deb и даже виртуальные диски.

Прежде чем запускать *Photorec*, создайте каталог, где он будет сохранять восстановленные файлы. По окончании работы этот каталог будет заполнен множеством файлов в разных форматах, со странными именами. Дело в том, что *Photorec* называет эти файлы по мере их нахождения, а разбираться в них предоставляет вам.

➤ **Параметры в меню Rescatux должны помочь вам исправить загрузчик, но если они не работают, всегда можно вернуться к инструменту Boot-Repair.**



## Когда ничего не получается...

Хотя Rescatux скорее поможет вам, чем нет, у него есть свои ограничения. Если вы всё сделали по этому учебнику и использовали должную функцию Rescapp, но по-прежнему не можете восстановиться после сбоя, то пришло время положиться на мудрость старейшин.

Rescatux — лютой борзописец и создает файлы журнала практически для всех задач. В разделе

Support вверху утилиты Rescapp перечислены параметры, которые могут пригодиться, когда не удается самостоятельно устранить проблему. Функция Show журнала открывает папку, где хранятся все журналы; вы можете их просмотреть, чтобы выявить точную причину сбоя задачи.

Если вы не в состоянии отыскать решение, можно использовать функцию Share log, чтобы

скопировать содержимое выбранного файла журнала на [pastebin.com](https://pastebin.com). Скопируйте URL и поделитесь им с другими — либо на соответствующих форумах инструментов, либо на их IRC-канале. Вы также можете использовать функцию Chat для запуска *xchat* и входа в IRC-канал `#rescatux`, где другие пользователи могут указать подходящий способ решения вашей проблемы.

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

Вдобавок, несмотря на то, что *Photorec* — утилита командной строки, она разбивает процесс восстановления файлов на этапы, подобно мастеру. При запуске этого инструмента отобразятся все жесткие диски и подключенные съемные устройства, включая все подключенные USB-накопители. Чтобы продолжить, выберите диск с потерянными файлами. Если диск содержит несколько разделов, *Photorec* отобразит все разделы и позволит выбрать тот, где размещались потерянные файлы. Затем инструмент должен узнать тип файловой системы, в которой хранятся ваши файлы. Есть только два варианта. Выберите опцию `[ext2/ext3]`, если удаленный файл находится внутри дистрибутива Linux. Параметр `[Other]` будет искать файлы, созданные в FAT/NTFS/HFS+ или любой другой файловой системе. Затем вам придется решить, будете ли вы отыскивать удаленные файлы внутри освобожденного пространства или на всем разделе. И последний шаг — указать *Photorec* созданную вами папку для хранения всех восстановленных файлов.

Вооружившись этой информацией, *Photorec* приступит к работе, на что может потребоваться довольно много времени, в зависимости от объема раздела, над которым он трудится. Все найденные файлы сохраняются в указанной вами папке. Заглянув

в папку назначения, вы увидите несколько подпапок с именами `recup_dir.1`, `recup_dir.2` и т.д. Восстановленные файлы сохраняются в них. Ручная сортировка этих файлов может занять вечность. Можно выполнить базовую сортировку из командной строки, чтобы лучше организовать файлы: например, скомпановать `mv ~/recovered/recup_dir.*/*.*.jpg ~/all-recovered-images`, чтобы переместить все файлы JPG из всех восстановленных папок в папку `all-recovered-images`.

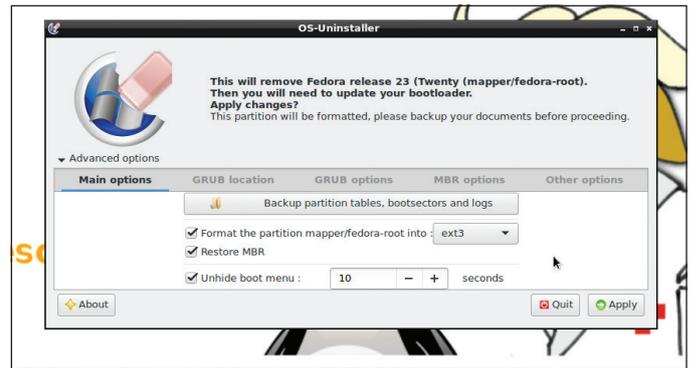
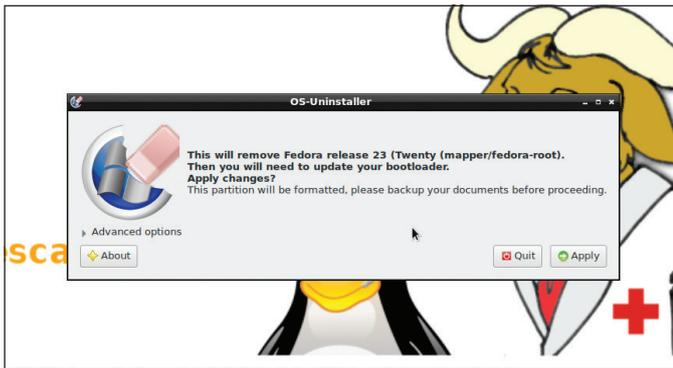
Можно также отсортировать файлы по размеру. Это очень полезно, особенно при восстановлении изображений. В дополнение к восстановлению самого изображения, *Photorec* также восстановит его миниатюры, которые будут иметь такое же расширение. Команда `find ~/all-recovered-images/ -name "*.jpg" -size -10k | xargs -i mv {} ~/thumbnails` перемещает все изображения размером менее 10 КБ из папки `all-recovered-images`.

Как видите, *Rescatux* — чрезвычайно полезный дистрибутив, способный помочь вам преодолеть сложную ситуацию. Хотя огромная работа выполняется мощными инструментами и утилитами командной строки с открытым исходным кодом, *Rescatux* делает их доступными для неопытных пользователей благодаря собственной утилите *Rescapp*, управляемой через меню. **LXF**

### Скорая помощь

Вместо того, чтобы тратить время на сортировку всех файлов, восстановленных *PhotoRec*, вы можете велеть инструменту искать и идентифицировать только определенные типы файлов.

## Управляйте своими ОС из OS Uninstaller

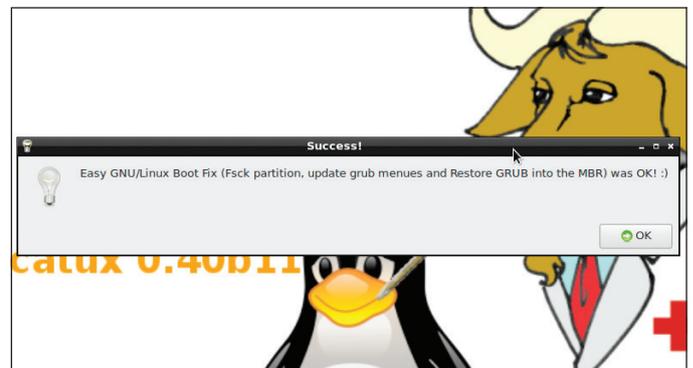
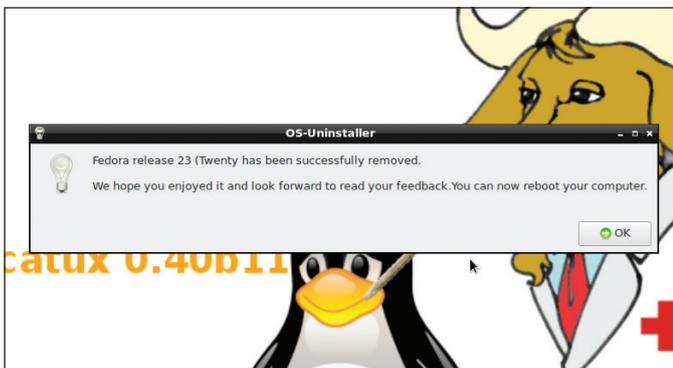


### 1 Запустите инструмент

Запустите утилиту *Rescapp* и прокрутите экран вниз до раздела Expert Tools. Нажмите кнопку OS Uninstaller, чтобы запустить этот инструмент. Он выполнит проверку ваших дисков и задаст кое-какие вопросы — например, используете ли вы RAID на диске. Инструмент завершит работу, когда обнаружит на одном из разделов диска каталог `/boot`.

### 2 Резервное копирование загрузчика

Приложение отображает список операционных систем и просит вас выбрать ту, которую вы хотите удалить. Вам также предложат создать резервную копию таблицы разделов и загрузочного сектора. Разверните выпадающее меню *Advanced options* и выберите пункт *Backup partition tables, bootsectors and logs*, чтобы указать место для их хранения.



### 3 Удалите ОС

Резервные копии разделов и загрузочного сектора пригодятся на случай, если удаление ОС повредит эти критические области, и их потом придется исправлять. После их резервного копирования со всеми данными перейдите к удалению ОС. Инструмент удалит все следы выбранного дистрибутива или ОС.

### 4 Обновите загрузчик

Скорее всего, у вас на этом компьютере есть другие дистрибутивы или операционные системы, помимо той, которую вы только что удалили. После удаления ОС нужно также и загрузчик уведомить об этом изменении. Для обновления загрузчика воспользуйтесь функцией Easy GNU/Linux Boot Fix.

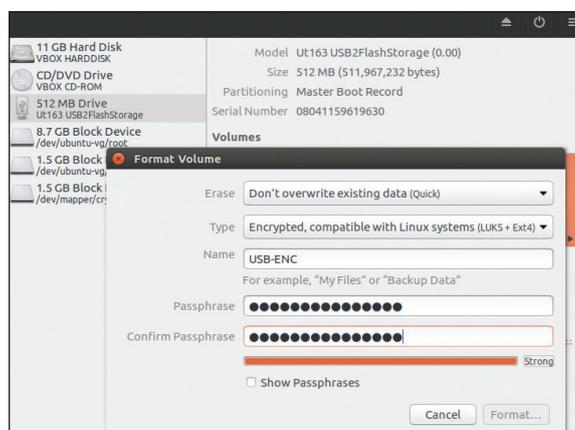
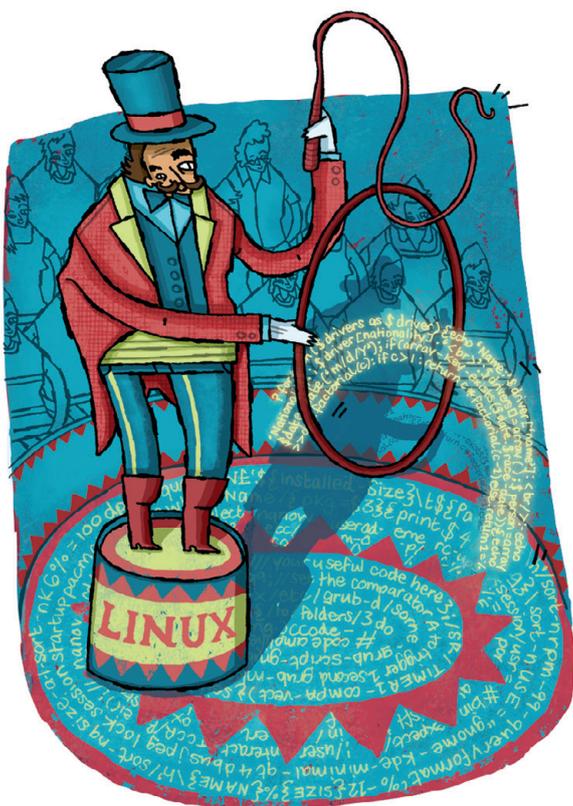
# LUKS: Шифруй и защитай

Нейт Дрейк исследует, как использовать инструменты, уже встроенные в вашу установку Linux, для шифрования файлов и защиты устройств.



Наш эксперт

**Нейт Дрейк** — журналист-фрилансер в области технологий, специализирующийся на кибербезопасности и ретро-технологиях. Он использует LUKS, чтобы спрятать свою карту сокровищ капитана Кидда.



» **Disk Utility** в Ubuntu может форматировать диск как том LUKS: просто введите свой пароль и нажмите **Format**.

На первый взгляд может показаться, что LUKS весело мурлычет под пресловутым капотом вашего компьютера и дальше уже беспокоиться не о чем. Наш урок предназначен для более любопытных читателей. Мы исследуем встроенные программы шифрования в Linux, и вы можете изменять их в соответствии со своими потребностями — допустим, выбрать собственный шифр или добавить к тому несколько паролей.

Наш урок сосредоточен на *Cryptsetup* — приложении командной строки, которое в большинстве дистрибутивов Linux предустановлено. Оно функционирует как интерфейс для модуля dm-crypt, позволяя быстро и легко шифровать диски. На простейшем уровне можно использовать *Cryptsetup* для создания простых томов, защищенных только паролем (см. врезку справа); но надо указывать все опции шифрования при каждом получении доступа к устройству, и, как назло, изменить пароль не так-то легко.

К счастью, *Cryptsetup* содержит полезные инструменты для создания и изменения зашифрованных томов LUKS. Помимо определения универсального стандарта шифрования для жестких дисков, тома LUKS содержат заголовки, указывающие шифр и хэш, а также уникальную 256-битную «присыпку», она же «приправа [salt, соль]», которая добавляется к паролю для обеспечения большей безопасности. Заголовок также содержит зашифрованный мастер-ключ, используемый для разблокировки контейнера. Это означает, что вы можете изменить пароль для доступа к мастер-ключу или добавить другие пароли.

Перед началом вашего путешествия по миру LUKS мы настоятельно рекомендуем настроить виртуальную машину или, как минимум, сделать резервную копию ваших данных. После этого откройте программу управления диском, чтобы просмотреть все зашифрованные LUKS-устройства, например, жесткий диск.

**В**се убежденные сединами, но при этом озабоченные конфиденциальностью пользователи Ubuntu, возможно, помнят, что в былые времена обеспечение безопасности было далеко не простым. Процесс предполагал либо высший пилотаж в командной строке, либо использование другого установочного CD для шифрования вашей установки Ubuntu.

Всё изменилось. Когда вы вставляете DVD Ubuntu и его разновидности, графический установщик предлагает вам опцию шифрования системы с помощью LVM (Logical Volume Manager), которая управляет дисками в LUKS (Linux Unified Key Setup), это стандарт шифрования Linux Hard Drive [Жесткого диска Linux].

Для большинства пользователей это прекрасный способ сэкономить время и защитить систему, не вникая во всякие туманные акронимы. По завершении установки системы ваш диск, за исключением раздела `/boot`, шифруется, и его можно разблокировать только через пароль. После входа большинство встроенных менеджеров дисков даже позволяют зашифровать внешний диск с помощью LUKS, чтобы он требовал пароль при каждом его подключении. Те же программы часто предлагают опцию изменения пароля для уже зашифрованных дисков.

Скорая помощь



Для полного перечня опций *Cryptsetup* посетите ЧАВО (FAQ) на <https://gitlab.com/cryptsetup/cryptsetup/wikis/FrequentlyAskedQuestions>.

```
nate@nate-VirtualBox:~$ sudo luksDump /dev/sdb1
Version: 1
Cipher name: aes
Cipher mode: xts-plain64
Hash spec: sha256
Payload offset: 4096
MK bits: 512
MK digest: ce ef 9c 4a 09 37 2b 10 00 34 e2 a8 8e 55 50 fe 44 4c f1 17
MK salt: dd ce 36 26 bc 03 d9 f7 08 f0 e4 91 af 55 28 08
63 88 d2 3d 85 a6 60 77 e6 0e e4 8e 6b 52 8f f4
MK iterations: 94500
UUID: 7dd446a7-45d2-485d-86d3-325f039c1a13

Key Slot 0: ENABLED
Iterations: 739982
Salt: f9 7a d2 09 34 f3 8a d9 df 4f ff f4 16 63 d4 1b
79 81 9a b0 36 2d e7 9b ca e3 a6 a0 44 da 98 2b
Key material offset: 8
AF stripes: 4000
Key Slot 1: DISABLED
Key Slot 2: DISABLED
Key Slot 3: DISABLED
Key Slot 4: DISABLED
Key Slot 5: DISABLED
```

➤ **luksDump** предоставит информацию о заголовке. Здесь используется 256-битное шифрование AES, разблокируемое одним паролем.

Вы увидите зашифрованный том LUKS, который появится в качестве отдельного устройства, например, `/dev/sda5`. Копнуть глубже поможет команда `luksDump` в терминале:

```
sudo cryptsetup luksDump /dev/sda5
```

## Храните ваши заголовки

После запуска команды `luksDump`, просто открывающей заголовок тома LUKS, первое, что вы видите — это опции шифрования по умолчанию, использованные, когда вы давали команду Ubuntu зашифровать для вас диск с помощью установщика или утилиты диска. В случае с шифрованием системы Ubuntu использует 256-битный AES с режимом XTS Block Cipher.

Также приводится «приправа» мастер-ключа, которая значительно повышает безопасность вашего тома LUKS. Без нее не будет никакой возможности извлечь ваши данные, даже с правильным паролем. По этой причине `Cryptsetup` позволяет создавать резервные копии ваших заголовков тома LUKS и восстанавливать их, поскольку если заголовок поврежден, извлечь ваши зашифрованные данные нельзя.

Чтобы сделать резервную копию вашего заголовка LUKS, откройте терминал и запустите команду `cryptsetup luksHeaderBackup <устройство> --header-backup-file <имя_файла>` — например, `sudo cryptsetup luksHeaderBackup /dev/sdb1 --header-backup-file`

`/home/nate/Desktop/backup1`. Для восстановления заголовка просто запустите ту же самую команду, заменив `luksHeaderBackup` на `luksHeaderRestore`.

Когда вы благополучно сделаете резервную копию вашего заголовка, запустите вышеупомянутую команду `luksDump` снова и прокрутите вывод вниз до раздела `Key Slots`. По умолчанию здесь восемь слотов ключей, пронумерованных от 0 до 7. Первый слот ключа (0) включается паролем, выбранным при изначальном шифровании тома. Если вы собираетесь выдать доступ к вашему тому LUKS кому-нибудь еще, можете добавить пароль или файл ключа, используя команду `Cryptsetup luksAddKey`, например, `sudo cryptsetup luksAddKey /dev/sda5`. Вам понадобится ввести существующий пароль. Новый пароль `Cryptsetup` попросит вас ввести дважды.

Чтобы удалить пароль из тома LUKS, воспользуйтесь опцией `luksRemoveKey`, например, `sudo cryptsetup luksRemoveKey /dev/sda5`. `Cryptsetup` попросит вас ввести пароль, который вы хотите удалить. Будьте осторожны с этой функцией, потому что если вы удалите все слоты ключей для контейнера LUKS, вы не сможете получить к нему доступ снова.

Kali Linux очень хорошо применяет эту функцию в патче стирания [nuke patch] для `Cryptsetup`, который при введении определенного пароля стирает все слоты ключей в заголовке LUKS системы (подробнее об этом см. на [www.kali.org/tutorials/nuke-kali-linux-luks](http://www.kali.org/tutorials/nuke-kali-linux-luks)). Это делается для защиты данных вашего устройства от кражи или захвата; но обязательно скопируйте куда-нибудь ваш заголовок, если решите использовать этот патч, иначе вы и сами не сможете получить доступ к своему компьютеру!

## Продвигая ваш LUKS

Хотя у `Cryptsetup` есть отличные опции по умолчанию для починки, вы, возможно, захотите получить большую степень контроля над своими данными. Во-первых, если вы не хотите шифровать весь том или диск, можно создать контейнер файлов LUKS, разблокируемый только по паролю. На стр. 70 вы увидите инструкции, как это сделать.

Если при создании своего контейнера вы используете опцию `luksFormat` саму по себе, то `Cryptsetup` будет применять опции шифрования по умолчанию. Однако, если вам так больше нравится, вы можете указать их сами. Например:



Заголовок из контейнера LUKS можно удалить, чтобы исключить возможность извлечения данных. Используйте для этого `head -c 1052672 /dev/zero > /dev/<your-luksdevice>.sync`

## Хороший LUKS или простенький?

Контейнеры LUKS не предпринимают никаких попыток скрыть свою натуру, и шифруют диск именно таким образом, вы делаете очевидным тот факт, что вам есть что скрывать. К счастью, вы можете использовать `Cryptsetup` в простом режиме [plain mode], в котором вас не выдадут никакие заголовки тома. Для всех, кто заглянет на диск, он покажется просто заполненным случайными данными.

Сначала определите местонахождение выбранного вами раздела или подключите USB-брелок, откройте терминал и очистите диск с помощью `dd`: например, `sudo dd if=/dev/urandom of=/dev/sdb1 bs=1M`. Основная команда для использования `Cryptsetup` в простом режиме — `cryptsetup <options> open --type plain <device> <name>`. Уделите некоторое время на ознакомление с главной страницей `Cryptsetup`, чтобы посмотреть, какие опции вам подходят, например:

```
sudo cryptsetup --type plain --cipher=serpent-xts-plain64 --hash=sha512 --key-size=256 --offset=123
```

```
open /dev/sdb1 plain3.
```

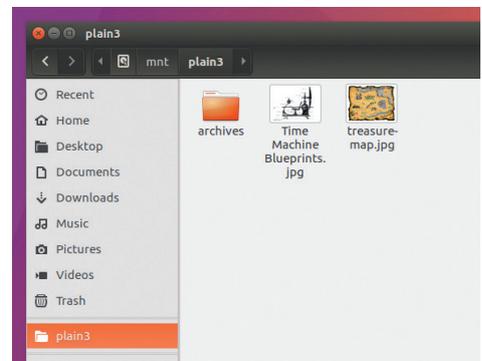
Эта команда должна вводиться совершенно одинаково каждый раз, когда вы хотите получить доступ к устройству. Если вы не хотите записывать ее, запомните величину смещения.

`Cryptsetup` попросит вас выбрать пароль. Убедитесь, что у него достаточно высокая степень энтропии, поскольку приправа здесь не используется (см. выше). Создайте файловую систему с помощью

```
sudo mkfs -t ext4 /dev/mapper/plain3
```

Потом создайте точку монтирования, запустив `sudo mkdir /mnt/plain3`, и смонтируйте устройство с помощью `sudo mount -t ext4 /dev/mapper/plain3 /mnt/plain3`. Теперь вы можете записывать ваши секретные файлы на `/mnt/plain3`. Закончив, запустите `sudo umount /dev/mapper/plain3`, а затем `sudo cryptsetup close plain3`.

Вы можете снова получить доступ к зашифрованному разделу, запустив опции `cryptsetup` и `mount`.



➤ Если ваша система такое поддерживает, добавьте смонтированное размещение для вашего тома с простым шифрованием в закладки менеджера файлов, для быстрого доступа.

```
sudo cryptsetup -y --cipher serpent-cbc-plain --key-size 256
--hash sha512 luksFormat test1
```

В этом примере мастер-ключ для контейнера test1 будет зашифрован с помощью алгоритма шифрования Serpent, который был финалистом в соревновании AES (Advanced Encryption Standard). Хэш SHA512 используется вместо SHA256.

*Cryptsetup* также поддерживает использование файлов ключей вместо паролей. Это особенно удобно при использовании простого режима, потому что, вероятнее всего, степень энтропии у файлов выше, чем у паролей. Для контейнеров LUKS можно добавить файл ключа, просто указав расположение файла после `luksAddKey`, например:

```
sudo cryptsetup luksAddKey /dev/sdb1 /home/nate/Pictures/kitten.jpg
```

Введите любой существующий пароль для добавления файла в слот ключа. Чтобы открыть контейнер LUKS, используя файл ключа, просто укажите путь с помощью опции

```
--key-file, например:
sudo cryptsetup luksOpen /dev/sdb1 vol1 --key-file /home/nate/Pictures/kitten.jpg
```

## Без понятия о LUKS

В приведенном выше примере в качестве существующего двоичного файла был использован котенок из Wikimedia. В реальности это

обеспечит очень невысокую степень безопасности, потому что любой, у кого есть копия файла, сможет открыть ваш контейнер LUKS. Создайте произвольный файл ключа, используя случайные данные, или используйте файл, доступ к которому есть только у вас, например, сделанную вами фотографию.

Если при создании точки монтирования или форматирования дисков вы использовали `sudo`, может оказаться, что вы сперва не сможете изменить файлы в зашифрованном контейнере, поскольку они принадлежат пользователю `root`. Запустите в терминале `sudo chmod -R 777 /mnt/<ваша_точка_монтирования>`, чтобы исправить это.

В отличие от сторонних программ вроде *Veracrypt*, *Cryptsetup* не может шифровать тома с помощью каскадов шифров, таких как AESTwofish-Serpent. Использование любого из этих алгоритмов обеспечит адекватную степень безопасности, но особо параноидальные при желании могут создать контейнеры LUKS один внутри другого. Просто повторите шаги учебника, используя разные опции шифрования для каждого контейнера.

Помните, что надежность вашего контейнера в немалой степени определяется количеством энтропии в вашем пароле. Если у вас хорошая память, попробуйте использовать для создания пароля *DiceWare* (<http://world.std.com/~reinhold/diceware.html>). Для читателей из Великобритании рекомендуем список слов Бейля, потому что он использует британское правописание и избегает американизмов.

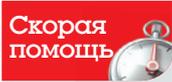
Если вам по душе удобство LUKS для смены паролей, но не хочется демонстрировать, что вы шифруете данные, можете создать контейнер LUKS с отделенным заголовком. Преимуществом здесь будет то, что сам по себе контейнер выглядит как случайные данные, обеспечивая вам до некоторой степени правдоподобное отрицание. Для этого при создании тома LUKS укажите место хранения заголовка опцией `--header`. Например:

```
sudo cryptsetup luksFormat /dev/sda --header /dev/sdb
--align-payload=0
```

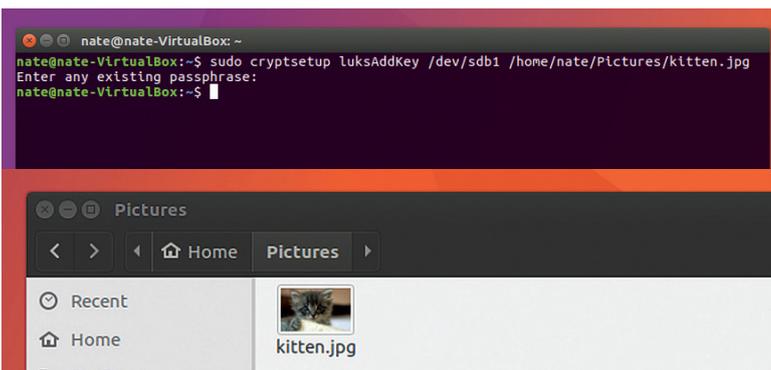
При монтировании томов LUKS укажите местоположение заголовков таким же образом:

```
sudo cryptsetup luksOpen /dev/sda --header /dev/sdb Luks
```

Однако позаботьтесь зашифровать весь диск или раздел, поскольку вряд ли вы назовете правдоподобную причину наличия особо крупных файлов случайных данных в своей системе! **LXF**

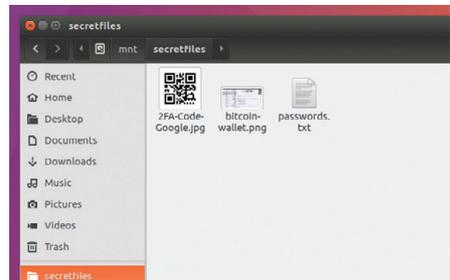
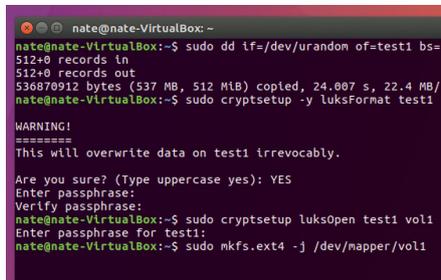
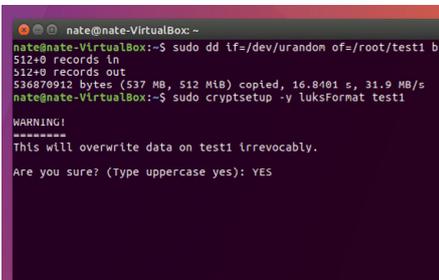


Создайте 512-КБ файл ключа из случайных данных с помощью `dd bs=1K count=512 if=/dev/urandom of=keyfile`



Введите любой действующий пароль для использования файла ключа, чтобы открыть тома LUKS. Используйте только файл, который вы создали сами.

## Создаем контейнеры файлов, зашифрованные в LUKS



### 1 Создайте контейнер LUKS

Откройте терминал и создайте файл из случайных данных с помощью `sudo dd if=/dev/urandom of=test1 bs=1M count=512`. Эта команда создает 512-МБ контейнер. Измените `bs` и `count` в соответствии с вашими нуждами. При желании вы можете также изменить имя с `test1` на нечто более осмысленное. Затем добавьте контейнеру заголовок LUKS с помощью `sudo cryptsetup -y luksFormat test1`. Введите `YES` и введите свой пароль дважды.

### 2 Смонтируйте свое устройство

Откройте свой новый контейнер по `sudo cryptsetup luksOpen test1 vol1`. Ваш файл откроется как локальное устройство обратной петли в `/dev/mapper/vol1`. Привойте вашему устройству файловую систему с помощью `sudo mkfs.ext4 -j /dev/mapper/vol1`. Затем создайте точку монтирования, например, `sudo mkdir /mnt/secretfiles`, и смонтируйте устройство: `sudo mount /dev/mapper/vol1 /mnt/secretfiles`. Откройте менеджер файлов и перейдите к своей точке монтирования.

### 3 Скопируйте файлы

Скопируйте свои секретные файлы в контейнер LUKS. На этом этапе вы можете захотеть сделать резервную копию заголовка тома LUKS (см. выше). Чтобы закрыть контейнер, запустите `sudo umount /mnt/secretfiles`, затем `sudo cryptsetup luksClose vol1`. При необходимости исправьте точку монтирования и имена тома. Чтобы открыть контейнер заново, следует запустить `sudo cryptsetup luksOpen test1 vol1`, затем `sudo mount /dev/mapper/vol1 /mnt/secretfiles`.

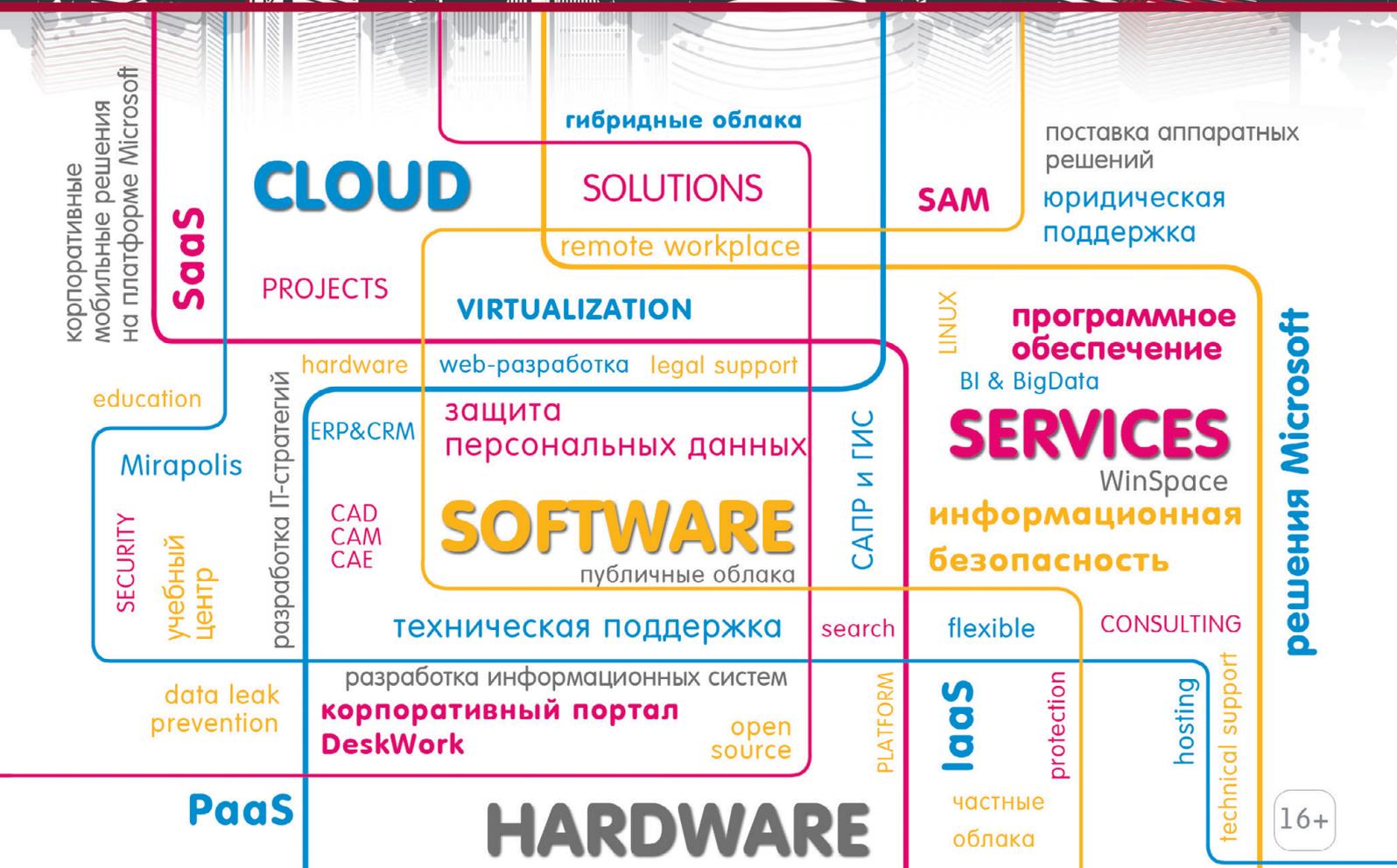
# softline®

Cloud Software Hardware Services

# 20+

Years in IT

## IT-архитектура вашего бизнеса



# Blockchains: Делаем деньги

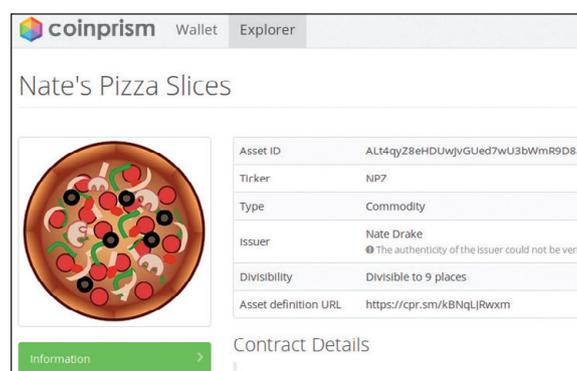
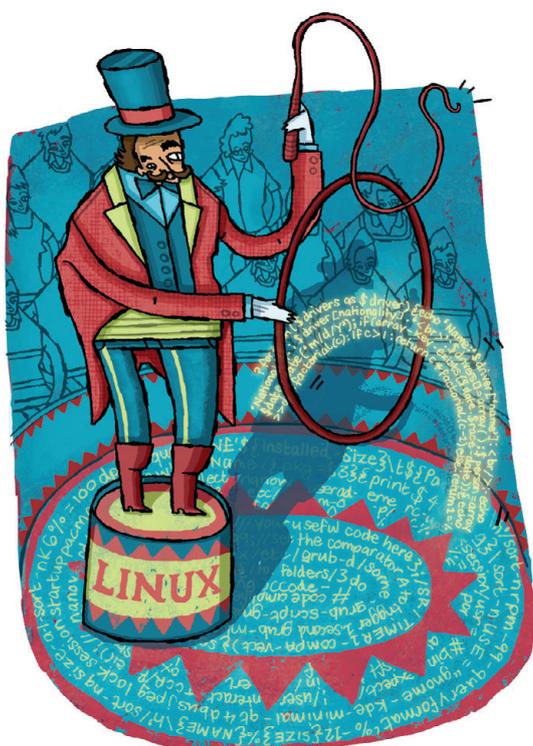
Нейт Дрейк демонстрирует, как применять Bitcoin Blockchain для временной отметки файлов, выпуска облигаций и даже создания собственной валюты.



Наш эксперт

## Нейт Дрейк

После создания валюты с обеспечением в виде пиццы Нейт уже никого не удивит, кроме самого себя, когда предлагает друзьям «акцию по пицце», заявляя, что он «замесил тесто».



» Блокчейн можно взнудать для создания разного вида цифровых активов через «окрашивание» биткойнов.

сообщения или файла. Если, например, вы только что настроили очередной великий роман, вы можете открыть терминал Linux и запустить команду `sha256sum mybook.epub`. Если бы вам затем надо было скопировать уникальный хэш, сгенерированный для этого файла и ввести его на сайте, например, <https://bitaddress.org>, вы бы увидели адрес Bitcoin, куда вы потом можете отправить свои грошики по курсу биткойнов. Всё, что потом надо сделать для удостоверения того, что вы владеете самой ранней версией файла — это повторить свои шаги и удостовериться, что вы отправляли монеты с этого адреса. Поскольку хэш уникален, существует постоянная запись того, что вы являетесь оригинальным автором, которая существует столько же, сколько существует блокчейн биткойнов.

Небесполезно знать, что вам придется оплатить транзакцию по каждому файлу или сообщению, сохраняемому в блокчейне, так что, возможно, вы предпочтете сервис вроде OpenTimestamps — там для бесплатной временной отметки файлов применяется структура данных, известная как дерево Меркла [Merkle tree].

## Чудеса с активами

Блокчейн биткойнов состоит из миллионов транзакций, обновляемых сериями блоков. Несколько лет назад хитроумные кодеры осознали, что, вставляя в транзакции небольшие количества данных, они могут создавать собственные псевдомонеты [color coins]. Их можно было бы использовать, чтобы отличить один биткойн от другого, позволяя хранить цифровые активы в блокчейне.

Представьте, например, что у вас ежемесячный клуб любителей пиццы для вас и ваших друзей. Используя псевдобиткойны, можно создавать и распределять цифровые единицы, которые ваши друзья могут употребить на оплату вкусного ломтика. Транзакции блокчейна неизменяемы: как только псевдомонету пустят в ход, это запишется, так что прожорливые друзья не смогут слопать больше того, что им причитается. Транзакции также криптографически

В мае 2016 г. австралиец по имени Крейг Райт [Craig Wright] заявил, что он является Сатоши Накамото [Satoshi Nakamoto], таинственным создателем биткойна [Bitcoin]. Его претензии в основном подкреплялись тем, что он мог связать свою личность с одной из самых первых произведенных транзакций биткойна.

Транзакции биткойнов управляются через блокчейн [Blockchain] — ком данных, который разбросан по тысячам серверов всего мира. Каждый пользователь биткойнов имеет частный ключ, используемый для подписи исходящих транзакций, а также открытый адрес Bitcoin, который можно сообщать другим для получения платежей. В большинстве случаев люди используют программы-кошельки для автоматического создания этих адресов, однако их без особых сложностей можно сделать самим.

Ведущий разработчик биткойна Гэвин Андерсен [Gavin Andersen] попросил Райта цифровым способом подписать сообщение "Gavin's favorite number is eleven [Любимое число Гэвина — одиннадцать]" с помощью частных ключей для первых 50 созданных биткойнов.

Хотя присяжные до сих пор совещаются насчет того, сказал ли Райт правду, его претензия обнаружила потрясающий новый способ использовать блокчейн биткойнов для временной отметки



Чтобы создать хэш фразы вместо файла, используйте 'echo', например, `echo "I am Keyser Soze" | sha256sum`.

## Извлечь максимум из OpenTimestamps

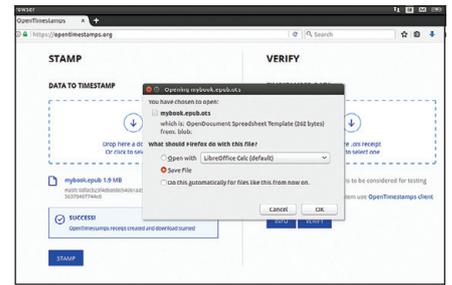
Хотя технически вы можете вручную самостоятельно сделать временную отметку файла, намного быстрее и проще использовать OpenTimeStamps (<https://opentimestamps.org>). Для начала посетите сайт и перетащите файл, который вы хотите пометить, в поле с именем Data to Timestamp [Данные для временной отметки], затем нажмите Stamp. Сайт создаст файл с расширением .ots. Сохраните этот файл на своем жестком диске. Вы можете удостовериться его в любое время, посетив сайт снова и перетащив его в раздел TimeStamped Data.

Сайт рекомендует использовать для удостоверения файлов opentimestamps-client, поскольку это более надежно. Для начала откройте Terminal и запустите `git clone https://github.com/opentimestamps/opentimestamps-client`, затем перейдите в директорию с помощью `cd opentimestamps-client`.

Установите необходимые зависимости посредством запуска `sudo apt-get install python3-pip`, затем `sudo pip3 install -r requirements.txt`. Запустите `ots stamp myfile.txt`, где `myfile.txt` — это выбранный вами файл для синхронизации с сервером OpenTimeStamps, и пометьте его.

Чтобы удостовериться подпись, вы должны сначала запустить на своем компьютере полный узел Bitcoin. Это может отнять немало дисковой памяти и полосы пропускания, поскольку по умолчанию он загружает весь блокчейн биткойнов на ваш компьютер. Однако вы можете использовать режим урезания [pruning]. См. <https://bitcoin.org/en/fullnode> для более подробной информации.

Когда ваш узел Bitcoin запустится, выполните `ots verify myfile.txt.ots` на файле подписи OpenTimeStamps, чтобы удостовериться ее.



➤ Просто перетащите файл в 'Data to TimeStamp', чтобы создать подпись. Любой может удостовериться подпись с помощью сайта или клиента opentimestamps.

удостоверяются, и вы можете точно знать, кто обналчил свою псевдомонету, и даже приблизительно когда.

Практически любой актив реального мира — акции, облигации и прочие товарные активы, допустим, драгметаллы — допускает управление посредством псевдомонет. Вы можете употребить язык скриптов клиента Bitcoin, чтобы приступить к этому, но подумайте о том, чтобы обратиться к CoinPrism. Этот удобный сайт позволяет выпускать и делиться активами с помощью блокчейна биткойнов, не вводя ни строки кода.

Прежде чем начать, подумайте, какой вид актива вы хотели бы создать и как им делиться. Например, если ваша компания стоит £1000, возможно, вы захотите продать 100 акций по £10 за каждую. Тогда вам понадобится создать 100 отдельных единиц пакета акций. Вы можете использовать CoinPrism, чтобы решить, будет ли актив делиться до определенного числа десятичных знаков. В случае с акциями это неприменимо, но при создании собственной валюты вы, возможно, решите рассмотреть вариант наличия дробной части.

Вам понадобится сумма более 0,001 BTC (на данный момент это чуть больше £2) на покрытие изначальных расходов по транзакциям для выпуска ваших псевдомонет. Вы можете или приобрести биткойны на бирже вроде BitStamp ([www.bitstamp.net](http://www.bitstamp.net)), или использовать сборщик биткойнов [Bitcoin faucet] для получения небольшого количества бесплатных монет.

### Денежные дела

Основное условие цифровых активов из псевдомонет заключается в том, что вы заключаете контракт с держателями, предоставляя им актив в реальном мире в обмен на их цифровой актив.

Между количеством Bitcoin, использованных для создания цифрового актива, и стоимостью самого актива не обязательно должна быть связь. Скажем, «Пицца-клуб Нейта» вручает по порции пиццы шестью едокам. Каждая из шести единиц имеет номинальную стоимость в биткойнах менее одного пенни.

Другой удобный способ понять эту концепцию — то, что сырье и труд, вовлеченные в создание каждого британского фунта, стоят менее десяти пенсов, однако монета доставит вам намного большую ценность, когда вы предъявите ее в местном магазине.

С учетом того, что актив может иметь любую ценность, которую вы ему присвоите, вы можете использовать псевдобиткойны весьма интересными способами, а не только выпуском акций или пиццы. Одним из примеров может быть создание местной валюты или платежных средств. Работает это так: откладывается определенное количество реальных денег и создается соответствующее

количество единиц валюты, которую можно потратить или возместить только в определенной местности, обычно по курсу 1:1.

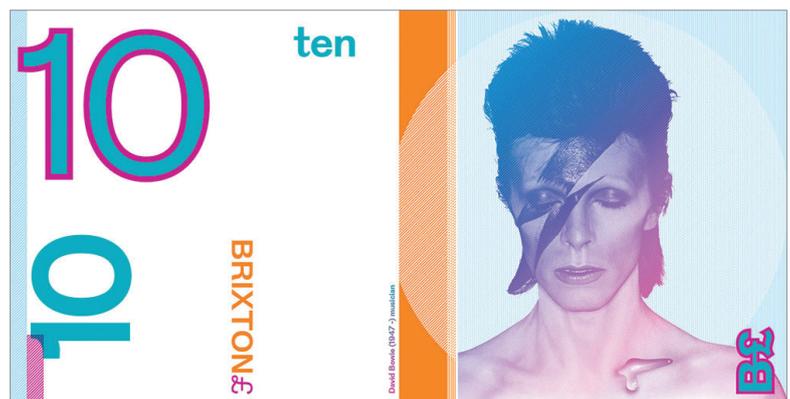
Например, представьте, что вы проживаете в городе Днищенске и хотите создать собственную днищенскую валюту под названием дноллары. Вначале вы откладываете некоторую сумму в реальных деньгах, например, 100 руб., и затем с помощью CoinPrism создаете 100 днолларов. Затем вы обещаете по первому требованию обменивать каждый дноллар на реальный рубль, так что горожане соглашаются принимать дноллар в местных магазинах.

В первый день вы можете отправиться в газетный киоск и купить на дноллар прессы. Потом хозяин киоска может использовать этот дноллар, чтобы побриться в местной парикмахерской. Парикмахер, в свою очередь, может подарить дноллар своей 10-летней дочке в качестве карманных денег, которыми она тут же оплатит проезд на автобусе. Автобусная компания затем может обратиться к вам, эмитенту, и обменять дноллар на рубль. Внимательные читатели должны заметить, что один и тот же дноллар перед возмещением своей номинальной стоимости поменял владельцев четыре раза.

Поскольку дноллар оборачивается несколько раз, в локальной экономике он способен творить чудеса. Австрийский город Вергль [Wörgl] в 1930-х гг. использовал похожую схему для снижения уровня безработицы, а также постройки домов и даже лыжного трамплина. Одним из преимуществ применения блокчейна биткойнов в тех же целях является то, что вы можете отслеживать перемещения единиц вашей местной валюты по мере их хождения по рукам.

**Скорая помощь**

Если вы используете CoinPrism для выпуска акций, вы также можете отправлять владельцам псевдомонет дивиденды. Просто нажмите Send Coins, затем Send Dividends слева.



➤ Брикстон, пригород Лондона, имеет свою валюту. Купюры доступны по тому же номиналу, что и фунты стерлингов, и принимаются в 80 местных учреждениях.

## Обналичка

Чтобы получить и погасить активы, вашим держателям активов понадобится совместимый кошелек. К счастью, на помощь опять приходит CoinPrism, с собственным кошельком для устройств Android, который может получать регулярные платежи в биткойнах, как и любых отправляемых вами активах. Кошелек CoinPrism отобразит, вместе с QR-кодом, отдельный адрес для получения активов, на который вы затем сможете отправить активы с вашего онлайн-кошелька. Любые активы, полученные вашими друзьями, отображаются отдельно в биткойнах.

Количество активов в кошельке не ограничено. Любой пользователь кошелька Android может также отправлять активы другим держателям совместимых с CoinPrism кошельков, хотя им может понадобиться несколько биткойнов, чтобы покрыть стоимость транзакции (см. ниже). Но если вы затеете выпускать новый цифровой актив, вы должны использовать настольную версию сайта.

Заставьте держателей ваших активов записать резервную копию своих кошельков, на случай потери их устройства Android. Как и с Bitcoin, как только актив отправляется на кошелек, вы не можете отозвать его за них.

## Псевдокризисы

Поскольку курс биткойна за последнее время подскочил, оплата транзакции при мелких сделках начинает раздражать. Другими словами, если вы присваиваете номинальную ценность активу в виде порции пиццы, стоимость его возмещения может быть

больше, чем стоимость самого актива. Самый простой обходной путь, если люди явятся за своим активом лично — вручную проверить кошелек на их мобильном устройстве. Вы можете избежать двойных выплат держателям, если через определенные промежутки времени будете выпускать новые активы. Например, одна порция «Пиццы Нейта» от августа 2017 г. может использоваться только в течение этого месяца этого года.

Если так не получается, вы можете решить не оплачивать стоимость транзакции при выпуске или погашении активов; но тогда у майнеров нет никакого стимула быстро обработать вашу транзакцию. CoinPrism имеет функцию Crowd Sale, позволяющую продавать активы за фиксированную сумму BTC, а также выбрать, кто будет оплачивать расходы по транзакции — покупатель или продавец. Нажмите Send Coins, а затем Crowd Sale в меню слева из вашей учетной записи CoinPrism для более подробной информации.

Хотя подделка транзакции — и, следовательно, собственность на активы — почти невозможно, блокчейн не защитит вас от нечестного эмитента. Если, например, человек создает больше виртуальных порций пиццы, чем он их испек, технология блокчейна не защитит пользователей от мошенничества. CoinPrism пытается снизить риск этого с помощью своего сайта (<https://coinprism.info>), предоставляющего информацию об активах и эмитентах.

Озабоченным безопасностью читателям может не понравиться необходимость доверить свои биткойны и активы стороннему сайту вроде CoinPrism. Но сайт заявляет, что он хранит только зашифрованную версию ваших частных ключей и требует от вас пароль для подписи всех транзакций. При первом входе вы также можете экспортировать частные ключи вашего кошелька на свой компьютер.

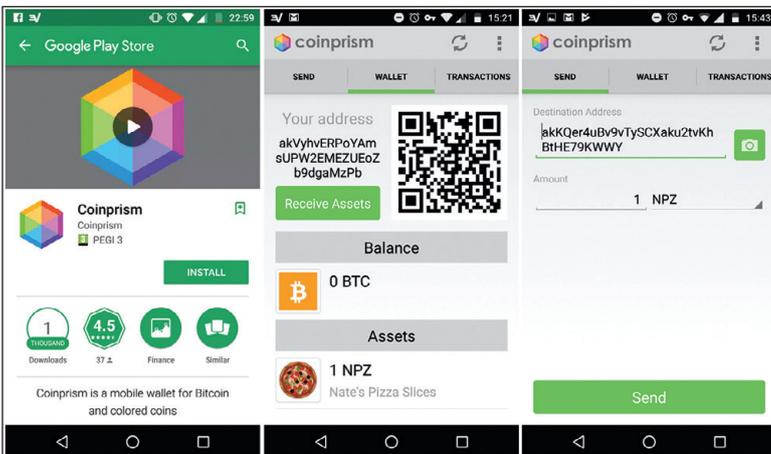
Если вам всё еще некомфортно, можете использовать кошелек Python с открытым кодом — *Colorcore* (<https://github.com/OpenAssets/colorcore>). *Colorcore* совместим с CoinPrism и протоколом OpenAssets и позволяет выпускать псевдомонеты из командной строки, не доверяясь третьим лицам. Но (не считая отсутствия дружелюбного к пользователю интерфейса), чтобы *Colorcore* работал нужным образом, вам также придется запустить на своем компьютере сервер Bitcoin Core, который скачает блокчейн биткойнов целиком, что на данный момент составляет примерно 125 ГБ.

Помните, что если у вас завалались активы, которыми вы больше не пользуетесь, вы можете их конвертировать и вернуть в свой обычный биткойновый кошелек. На сайте CoinPrism просто нажмите Send Coins, и затем Uncolor Coins слева.

Что бы еще вы ни делали — решив создать свои псевдомонеты, обязательно включите свое воображение, и не бойтесь исследовать и копировать другие кошельки и протоколы, если вам кажется, что они больше соответствуют вашим потребностям. **LXF**

### Скорая помощь

Приложения псевдомонет включают золотые слитки, порции бекона от свиньи и даже участки на Луне. Загляните в активы [www.coinprism.info](http://www.coinprism.info) для большего вдохновения.



➤ Установив кошелек Android CoinPrism, получайте и отправляйте активы. Приложение отображает отдельные адреса для Bitcoin и может служить кошельком биткойнов.

## Пригоршня протоколов

Один из самых старых и простых способов сделать псевдобиткойны, известный как EPOBC, заключается в том, чтобы изменить поле nSequence при отправке транзакций Bitcoin (обычно оно не используется). Поскольку это не изменяет размеры транзакции, отправка псевдомонет таким образом не дороже, чем обычная. Однако nSequence — всего лишь 32-битное целое число, что не позволяет отправлять больших количеств метаданных по выбранному вами активу. Вам также придется заплатить некое минимальное количество биткойнов за каждую транзакцию.

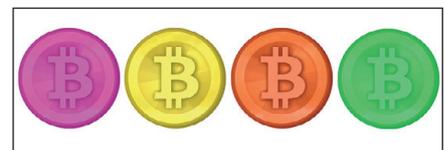
Протокол OpenAssets (<https://github.com/OpenAssets/open-assets-protocol>), используемый CoinPrism, старается усовершенствовать EPOBC посредством изменения кода операции OP\_RETURN в Bitcoin

Script. Метаданные хранятся в Blockchain и могут быть связаны с чем угодно в Сети, например, с серверами CoinPrism.

Реализация Colu Colored Coins (<http://colu.co>) также использует OP\_RETURN для хранения метаданных об активах Bitcoin, связывая их с публично доступными torrent-файлами. Это лучше, чем полагаться на любую компанию, поскольку torrent-файлы не уходят из сети, если эмитент вышел из дела. Другим преимуществом использования torrent-файлов является отсутствие ограничений на количество хранимых данных.

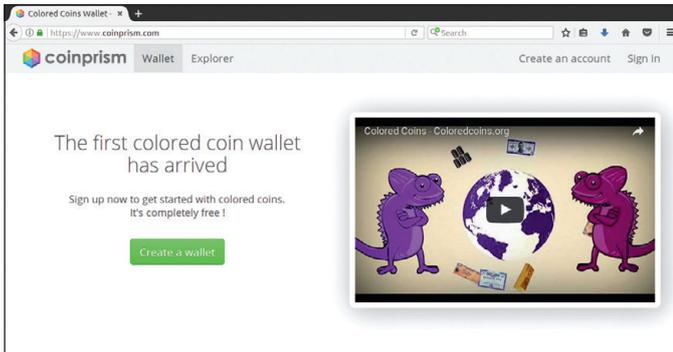
Colored Coins Protocol (<https://github.com/Colored-Coins>) позволяет эмитентам создавать правила для цифровых активов. Вы можете создать индивидуальные «умные контракты», уточняя комиссию,

срок действия, кому разрешается держать актив и можете ли вы, как эмитент, извлечь из него больше. Это может оказаться полезным в случаях, где актив имеет ценность только в определенное время для определенного человека, например, для того, кто берет в аренду ваш автомобиль.



➤ Потратьте немного времени, чтобы понять, какой именно метод псевдомонет лучше всего соответствует вашим потребностям.

## Чеканка ваших первых монет

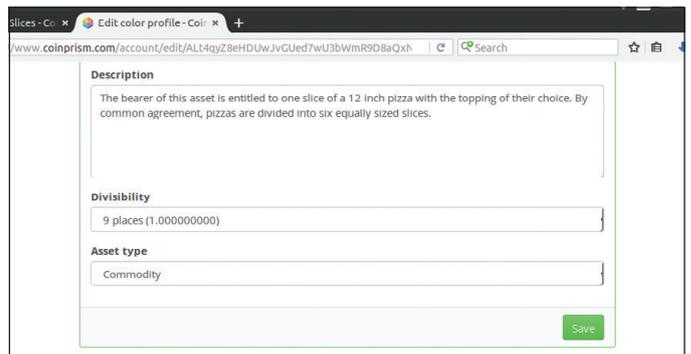
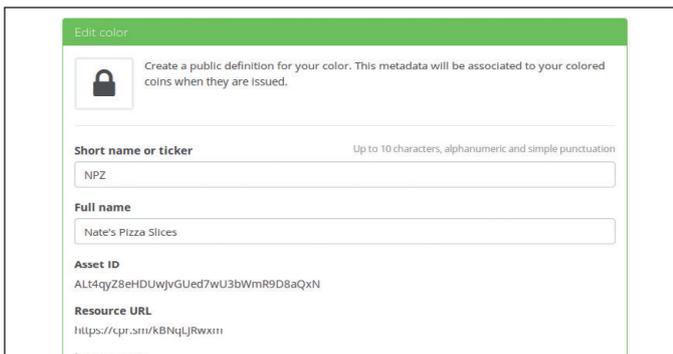


### 1 Зарегистрируйтесь на CoinPrism

Откройте ваш браузер и перейдите на <https://coinprism.com>. Нажмите Create a Wallet и введите желаемое имя пользователя, адрес электронной почты и пароль (как минимум из 10 символов). CoinPrism отправит вам письмо с вашим именем пользователя и адресом Bitcoin, где вы сможете получать платежи. Запишите его вместе со своим паролем, а затем вернитесь на <https://coinprism.com> и нажмите Sign in.

### 2 Добавьте адрес псевдомонеты

На странице Wallet Overview нажмите Addresses and Colors вверху справа, а затем на зеленую кнопку под названием New color. Введите для него имя. Если вы хотите использовать это с вашим web-кошельком, то оставьте Address Type в виде Regular, но вы сможете создать адрес холодного хранения/оборудования, если захотите. Снова введите свой пароль и нажмите Create.

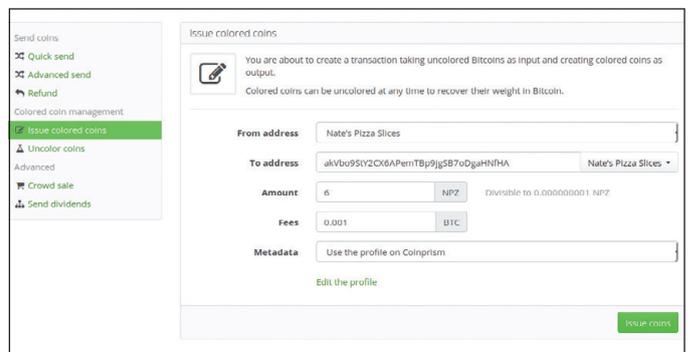
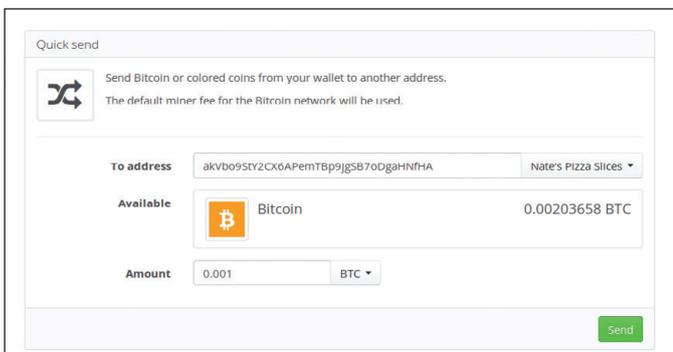


### 3 Милые активы

В следующем окне вы введете информацию о вашем активе. Введите в первом поле описание в стиле биржевого кода, например, NPZ. При желании можете также исправить Full Name. Под именем эмитента введите имя владельца актива. Запишите Asset URL. Вы можете использовать его, чтобы предоставить другим больше информации о вашем активе.

### 4 Описание и товар

Попробуйте создать подробное описание актива и добавить изображение по желанию. В делимости выберите '9 places': это даст вам больше гибкости при делении активов. Под Asset Type выберите правильный тип актива, например, ценные бумаги или облигации. Поскольку пицца — это еда, правильный тип здесь — товар. По завершении нажмите Save.



### 5 Адрес финансирования

Нажмите Send Bitcoins вверху, после чего выберите ваш актив из меню To. В поле количества отправьте сумму в BTC, которую вы собираетесь конвертировать. Это не обязательно должно быть большое число, но для ускорения транзакции попробуйте включить минимум 0,001 BTC, чтобы покрыть комиссию. Нажмите Send, заново введите свой пароль и нажмите Submit.

### 6 Выпуск псевдомонет

Снова нажмите Send Coins из меню наверху, а затем — на Issue color coins слева. Под Amount выберите количество единиц актива для выпуска: например, шесть кусков пиццы. Можно установить комиссию за транзакцию на более низкое число, но это будет означать, что на создание вашего актива уйдет больше времени. И, наконец, нажмите Issue Coins. Ваш новый актив появится в вашем профиле CoinPrism.

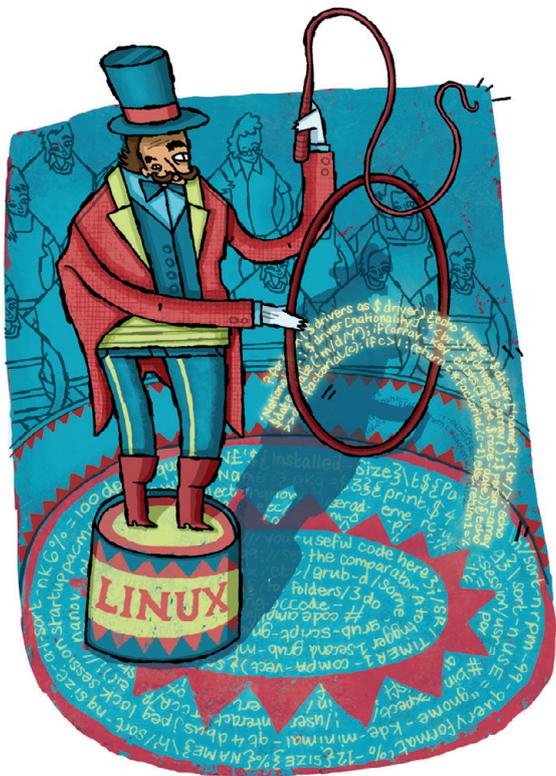
# ZFS: Как с ней начать

**Михалис Цукалос** показывает, как добавить разделы ZFS на компьютеры Linux и как создавать моментальные снимки на случай отката системы.



**Наш эксперт**

**Михалис Цукалос** — администратор UNIX, программист, администратор баз данных и математик, которому нравится писать статьи и учиться чему-то новому. В данный момент он пишет книгу под названием *Go Systems Programming*. Можете связаться с ним через [www.mtsoukalos.eu](http://www.mtsoukalos.eu) и @mactsouk.



```
mtsouk@LTng:~$ sudo lsblk -o NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL
[sudo] password for mtsouk:
NAME FSTYPE SIZE MOUNTPOINT LABEL
sda   20.3G
├─sda1 ext4   16.3G /
├─sda2             1K
├─sda5 swap     4G [SWAP]
sdb   3G
sdc   3G
sdd   3G
sr0   1024M
mtsouk@LTng:~$ sudo fdisk -l
Disk /dev/sda: 20.3 GiB, 21771468800 bytes, 42522400 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x23c049d4

Device Boot      Start         End      Sectors  Size Id Type
/dev/sda1 *          2048     34134015   34131968  16.3G 83 Linux
/dev/sda2                34136062  42520575   8384514    4G  5 Extended
/dev/sda5                34136064  42520575   8384512    4G  82 Linux swap /
```

» Вот вывод от утилит *lsblk*, *fdisk* и *parted*, которые позволяют выяснить, какие диски и разделы доступны на вашем Linux-компьютере.

никуда не денутся! Идеальной файловой системы не существует, и каждая новая пытается улучшить определенные области предыдущих файловых систем, что в основном касается эффективности и безопасности данных.

«За сценой» ZFS следует определенной философии. Для управления физическими дисками она использует пулы хранения, вместо того, чтобы делить физические диски на разделы, и предлагает поддержку моментальных снимков. Моментальный снимок ZFS — это копия файловой системы ZFS, доступная только на чтение, то есть ее нельзя изменить. Моментальные снимки ZFS полезны для того, чтобы вернуться назад во времени и восстановить удаленные или измененные файлы. Если вы когда-либо форматировали новый жесткий диск на компьютере с Linux, вы должны быть в курсе того, что перед использованием новый жесткий диск надо разбить на разделы. Однако ZFS умеет работать с жесткими дисками, и не разбитыми на разделы. Использование всего жесткого диска — распространенная практика в конфигурациях RAID.

Является ли ZFS панацеей? Конечно, нет! ZFS лучше всего подходит для приложений, которые требуют безопасности данных, при условии наличия у вас множества жестких дисков и большого объема памяти.

Прежде чем мы продолжим наш урок: помните, что есть два типа RAID: программный и аппаратный. Это руководство относится только к программному RAID, потому что аппаратный RAID

**Z**FS — это 128-битная файловая система, способная справиться с зеттабайтами данных — один зеттабайт равен миллиону терабайт! ZFS была разработана Sun Microsystems как файловая система, предоставляющая защиту от потери данных и поддержку очень больших разделов, и изначально была проприетарным проектом. Однако это изменилось в 2005 г., когда код ZFS был открыт.

Мы рассмотрим, как добавить раздел ZFS в существующую систему Linux, использующую ext4 в качестве корневой файловой системы *root (/)*, как удалить разделы ZFS и как создавать и использовать моментальные снимки ZFS.

## Про ZFS

Первый вопрос, который может у вас возникнуть — зачем вам еще одна файловая система? За которым очень быстро следует логичный вопрос: что дурного в файловых системах ext2, ext3 и ext4? Ну, совершенства в мире нет, и файловые системы тоже от этого

## Скорая помощь

Можно ли конвертировать файловую систему ext4 в ZFS? Короткий ответ — нет: вначале надо создать новый раздел ZFS, а затем скопировать существующие данные в него. Так что легких путей не будет!

## Каковы требования к аппаратуре?

В принципе, чем больше у вас ОЗУ и чем быстрее процессор, тем выше производительность ZFS, что, конечно, истинно практически для всего, касающегося компьютера!

Однако, по общему правилу, понадобится около 1 Гб ОЗУ на каждый терабайт имеющегося у вас хранилища, и начинать следует с системы Linux при объеме ОЗУ как минимум 4 Гб ECC. Вдо-

бавок рекомендуется использовать скоростные жесткие диски.

Наконец, чтобы создать раздел RAID 5 ZFS, понадобится как минимум три физических жестких диска. Но если вы хотите просто протестировать ZFS, можно использовать виртуальную машину и создать сколько угодно виртуальных дисков для тестирования. Пожалуйста, никогда не используйте

ZFS с аппаратным контроллером RAID, поскольку это не будет работать.

В конечном же итоге всё зависит от того, быстро ли вам надо получать доступ к данным и долго ли вы готовы их ждать. Следовательно, вы можете использовать меньший объем ОЗУ на компьютере, используемом для тестирования — но никак не на рабочем сервере.

создается до установки операционной системы. Следовательно, ОС ничего не знает о его конфигурации.

Первое, что вам надо сделать — выяснить количество жестких дисков на вашем компьютере Linux. Это простая задача, включающая выполнение одной из следующих команд:

```
$ sudo lsblk -o NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL
```

```
$ sudo fdisk -l
```

```
$ sudo parted -l
```

Рис. на стр. 76 показывает полный вывод первых двух команд, а также небольшую часть вывода последней команды. Все три утилиты могут выполнить вашу работу, и все согласны, что вышеупомянутый компьютер Linux имеет четыре диска: один из них в данный момент занят, а остальные три пока еще не форматированы, и следовательно, могут использоваться в установке ZFS.

Каждый из трех неиспользуемых жестких дисков имеет емкость 3 Гб. Имя занятого диска — `/dev/sda`, и он разбит на три раздела: `/dev/sda1`, `/dev/sda2` и `/dev/sda5`. Имена устройств трех свободных жестких дисков — `/dev/sdb`, `/dev/sdc` и `/dev/sdd`.

После выяснения доступных дисков на компьютере Linux будьте очень осторожны с остальными командами, поскольку можно полностью разрушить всю вашу систему Linux, попытавшись отформатировать или разбить на разделы не тот диск или раздел!

## Установка ZFS

Первым делом надо установить необходимое ПО на ваш компьютер Linux. В системе Ubuntu Linux это означает установку пакета ZFS и его зависимостей:

```
$ sudo apt install zfs
```

Команда `sudo zfs list` позволяет увидеть список файловых систем ZFS. Вдобавок она подтверждает, что с текущей установкой ZFS всё работает нормально. Поэтому выполнение предыдущей команды после новой установки ZFS создаст следующий вывод:

```
$ sudo zfs list
```

```
no datasets available [доступных данных нет]
```

## Администрируем ZFS

Скучным задачам нет конца... настало время выполнить некоторые основные задачи, относящиеся к ZFS, поскольку, в конце концов, если вы не можете создать раздел ZFS, зачем тогда устанавливать ZFS на ваш Linux-компьютер? Итак, этот раздел покажет, как создать раздел RAID 0. Поскольку в раздел RAID 0 войдут все три свободных жестких диска, общий размер созданного раздела будет около 9 Гб. Команда ZFS для создания раздела RAID 0 выглядит следующим образом:

```
$ sudo zpool create -f LXF /dev/sdb /dev/sdc /dev/sdd
```

Возможно, вас удивит, что одна команда способна создать новый раздел RAID 0, готовый к употреблению, поскольку он монтируется автоматически! Рис. справа показывает вывод предыдущей команды, а также вывод от команд `zfs list` и `zpool status`.

## Использование ZFS

Итак, свеже созданный раздел ZFS автоматически смонтирован и готов к сохранению на нем данных. Единственная проблема тут такая: его поведение по умолчанию означает, что только пользователь root имеет право хранить данные в разделе ZFS:

```
$ ls -dl /LXF/
```

```
drwxr-xr-x 2 root root 3 Jun 12 22:12 /LXF/
```

```
$ touch /LXF/aFile
```

```
touch: cannot touch '/LXF/aFile': Permission denied
```

Выполнение следующей команды `chown` разрешит пользователю 'mtsouk' использовать раздел LXF ZFS для хранения его данных:

```
$ sudo chown mtsouk:mtsouk /LXF
```

```
$ touch /LXF/aFile
```

```
$ ls -ld /LXF
```

```
drwxr-xr-x 2 mtsouk mtsouk 4 Jun 13 11:11 /LXF
```

```
$ ls -l /LXF
```

```
total 1
```

```
-rw-rw-r-- 1 mtsouk mtsouk 0 Jun 13 11:11 aFile
```

```
-rw-r--r-- 1 root root 0 Jun 12 22:12 doesntWork
```

Вывод следующей команды сообщит вам о смонтированных разделах ZFS, не требуя прав root:

```
$ mount | grep -i zfs
```

```
LXF on /LXF type zfs (rw,relatime,xattr,noacl)
```

```
LXF/docs on /docs type zfs (rw,relatime,xattr,noacl)
```

```
LXF/tutorials on /tutorials type zfs (rw,relatime,xattr,
```

```
noacl)
```



OpenZFS — это проект с открытым кодом, который объединяет компании и людей, использующих ZFS и желающих сделать его лучше. Вы можете узнать больше об OpenZFS, посетив <http://open-zfs.org>.

```
2. mtsouk@LTTng: ~ (ssh)
mtsouk@LTTng:~$ sudo zpool create -f LXF /dev/sdb /dev/sdc /dev/sdd
mtsouk@LTTng:~$ sudo zpool create -f LXF /dev/sdb /dev/sdc /dev/sdd
invalid vdev specification
the following errors must be manually repaired:
/dev/sdb1 is part of active pool 'LXF'
/dev/sdc1 is part of active pool 'LXF'
/dev/sdd1 is part of active pool 'LXF'
mtsouk@LTTng:~$ sudo zfs list
NAME      USED  AVAIL  REFER  MOUNTPOINT
LXF      56.5K 8.67G   19K    /LXF
mtsouk@LTTng:~$ sudo zpool status
pool: LXF
state: ONLINE
scan: none requested
config:

    NAME      STATE  READ  WRITE  CKSUM
    LXF       ONLINE  0     0     0
        sdb    ONLINE  0     0     0
        sdc    ONLINE  0     0     0
        sdd    ONLINE  0     0     0
```

» Применение команды `zpool create` для создания раздела RAID 0 ZFS, созданного из трех жестких дисков, а также применение и вывод команд `zpool status` и `zfs list`.

» Подпишитесь на печатную или электронную версии на [www.linuxformat.ru/subscribe!](http://www.linuxformat.ru/subscribe!)

## Моментальные снимки

Вы можете создать новый моментальный снимок ZFS LXF с именем snapshot1 с помощью команды `zfs snapshot` следующим образом:

```
$ sudo zfs snapshot LXF@snapshot1
```

Доступные моментальные снимки ZFS вы найдете, скомандовав

```
$ sudo zfs list -t snapshot
```

Дополнительную информацию обо всех доступных моментальных снимках ZFS вы получите по команде `zpool list` с соответствующими опциями командной строки.

Командой `zfs destroy` можно уничтожить существующий моментальный снимок. Рис. внизу показывает вывод предыдущих команд, создавших разные моментальные снимки. Но на самом деле моментальные снимки используются для защиты ваших данных. В следующем сценарии представьте, что вы случайно удалили `/LXF/aFile` и хотите откатиться до предыдущей версии файловой системы с помощью моментального снимка. Введите следующее:

```
$ ls -l /LXF/aFile
-rw-rw-r-- 1 mtsouk mtsouk 0 Jun 13 11:11 /LXF/aFile
$ rm /LXF/aFile
$ touch aNewerFile
$ ls -l /LXF/
total 1
-rw-rw-r-- 1 mtsouk mtsouk 0 Jun 13 12:17 aNewerFile
```

```
-rw-r--r-- 1 root root 0 Jun 12 22:12 doesItWork
$ sudo zfs rollback LXF@snapshot2
$ ls -l /LXF/
total 1
-rw-rw-r-- 1 mtsouk mtsouk 0 Jun 13 11:11 aFile
-rw-r--r-- 1 root root 0 Jun 12 22:12 doesItWork
```

Хитрость в том, что файл `/LXF/aNewerFile`, созданный позднее моментального снимка `LXF@snapshot2`, после отката до `LXF@snapshot2` будет уничтожен.

Попытавшись использовать более старый моментальный снимок (`LXF@snapshot1`), вы увидите следующее сообщение об ошибке, информирующее вас о существовании более нового снимка:

```
$ sudo zfs rollback LXF@snapshot1
cannot rollback to 'LXF@snapshot1': more recent snapshots or
bookmarks exist
[откат невозможен: есть снимки и закладки новее]
use '-r' to force deletion of the following snapshots and bookmarks:
[используйте '-r' для принудительного удаления следующих
снимков и закладок]
LXF@snapshot2
```

## Удаление файловой системы ZFS

Теперь представьте, что вы используете файловую систему ZFS, но по какой-то причине она вам надоела, и вы решаете ее удалить. Этот раздел расскажет, как это сделать. И опять-таки, надо быть очень осторожным перед выполнением представленной команды, иначе за одно неправильное нажатие вы можете потерять все свои ценные данные!

Итак, если вы уверены в имени раздела ZFS, который хотите удалить, выполните следующую команду — с привилегиями администратора или используя утилиту `sudo`:

```
$ sudo zpool destroy LXF
```

И снова вам понадобится всего лишь одна команда, чтобы удалить весь раздел ZFS, что упрощает вашу жизнь... при условии, что вы случайно не удалите не тот раздел ZFS!

Результатом предыдущей команды будут три неиспользуемых жестких диска, которые вы можете употреблять по своему усмотрению. Следующий раздел расскажет, как использовать эти три жестких диска для создания файловой системы RAID 5 ZFS.

## Создание раздела RAID 5

Этот раздел будет использовать три свободных жестких диска из предыдущего раздела для создания установки RAID-Z — это улучшенная версия RAID 5, которая избавляет от известных

```
2. mtsouk@LTNg: ~ (ssh)
mtsouk@LTNg:~$ sudo zfs snapshot LXF/snapshot1
usage:
  snapshot|snap [-r] [-o property=value] ... <filesystem|volume>@<snap>

For the property list, run: zfs set|get

For the delegated permission list, run: zfs allow|unallow
mtsouk@LTNg:~$ sudo zfs snapshot LXF@snapshot1
mtsouk@LTNg:~$ sudo zfs snapshot LXF@snapshot2
mtsouk@LTNg:~$ sudo zfs snapshot LXF@snapshot2
cannot create snapshot 'LXF@snapshot2': dataset already exists
mtsouk@LTNg:~$ sudo zfs list -t snapshot
NAME                USED  AVAIL  REFER  MOUNTPOINT
LXF@snapshot1       0    -    19K    -
LXF@snapshot2       0    -    19K    -
LXF@snapshot1       0    -    19K    -
LXF@snapshot2       0    -    19K    -
LXF/docs@snapshot2  0    -    19K    -
mtsouk@LTNg:~$ zfs create LXF/tutorials -o mountpoint=/tutorials
```

➤ **Выполнение различных команд администрирования, относящихся к мгновенным снимкам ZFS, включая команды, создающие новые мгновенные снимки и удаляющие существующие.**

## Выясните свои уровни RAID

RAID означает Redundant Array of Independent Disks [Массив независимых дисков с избыточностью] — это технология, позволяющая хранить данные на нескольких жестких дисках, чтобы защитить ваши данные на случай, если один или несколько жестких дисков выдадут сбой.

Есть множество конфигураций RAID, включая RAID 0, RAID 1, RAID 5, RAID 6 и RAID 10, известный также как RAID 1+0.

RAID 0 не предлагает резерва, поскольку он просто позволяет рассматривать несколько дисков как один, что повышает производительность в большинстве операций, поскольку большинство из них

совершается на нескольких дисках параллельно. Но если один из дисков сломается, то, скорее всего, вы потеряете все свои данные. Вывод тот, что RAID 0 не стоит использовать для хранения критически важных данных.

В RAID 1 каждая пара дисков настроена как зеркальное отражение друг друга. А значит, если сломается один диск из двух, данных вы не потеряете; основное преимущество RAID 1 — его простота.

RAID 5 требует как минимум три диска и использует технологию под названием распределение данных по всем доступным жестким дискам. В результате, если ломается один диск, вы не теряете

данных. Однако, если одновременно сломается два или более диска, это означает потерю всех ваших данных.

RAID 6 похож на RAID 5, но добавляет еще один блок четности, что означает, что вы потеряете больше места на диске. Но если даже два диска в конфигурации RAID 6 сломаются одновременно, вы всё равно сохраните свои данные.

RAID 10 также называется линейкой зеркал. Предлагаю большую избыточность, чем RAID 5, он тратит еще больше места на диске. И чтобы создать конфигурацию RAID 10, нужны как минимум четыре жестких диска,

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

проблем с повреждением данных в RAID. Проще говоря, вы должны считать RAID-Z тем же, что и RAID 5, но без ошибок.

Раздел RAID-Z создается единственной командой:

```
$ sudo zpool create -f LXF raidz /dev/sdb /dev/sdc /dev/sdd
```

На сей раз объем созданного раздела будет намного меньше, чем в случае с RAID 0, и он будет менее подвержен потерям из-за поломки жесткого диска:

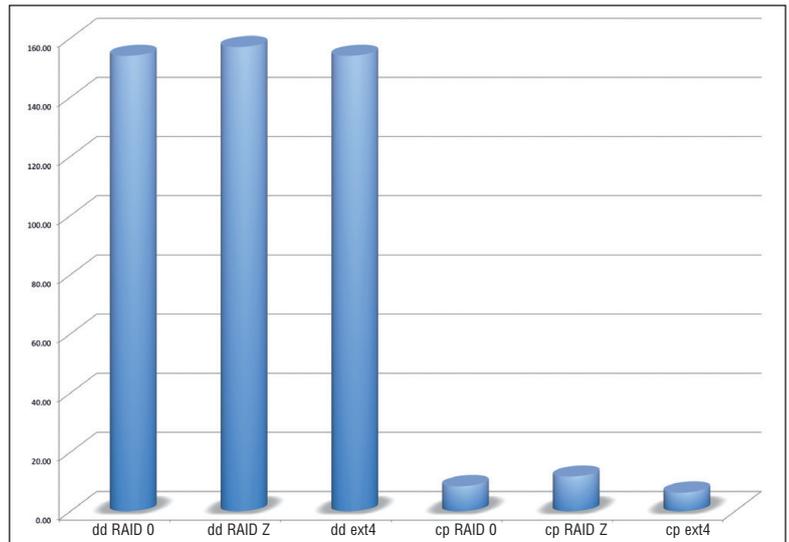
```
$ df -h | grep LXF
LXF 5.8G 0 5.8G 0% /LXF
```

Рис. внизу показывает вывод предыдущей команды, создавшей раздел RAID-Z, а также вывод команды `zpool destroy`, удалившей существующий раздел ZFS.

## Преимущества и недостатки

Значительное преимущество ZFS — она позволяет вам делать свою работу, не требуя редактировать какие-либо файлы конфигурации или перезапускать серверные процессы. В общем, ZFS использует относительно простой подход, делая то, что вам надо, одной командой. Кроме того, файловые системы ZFS полностью прозрачны для пользователей. Общее ощущение от использования ZFS — работать с ними одно удовольствие, и они облегчают жизнь администратора Linux, а ведь именно это и важно!

Однако у использования ZFS есть и недостатки, которые стоит учесть, когда вы будете рассматривать возможность применения этого формата файловой системы. Простые команды администрирования способны создать проблемы для новых пользователей, поскольку ошибочное выполнение команды может разрушить



► Это график значений поля `real` утилиты `time` при измерении выполнения команд `dd` и `cp` в `ext4` и ZFS.

существующие данные. Кроме того, подход ZFS является наиболее уместным, если вы имеете дело с большим количеством жестких дисков, так что это не самый лучший выбор для Linux-компьютеров с менее чем тремя жесткими дисками. И, наконец, для эффективной работы ZFS нужен приличный объем ОЗУ.

## Производительность

Как узнать, что всё работает хорошо? Просто: выполните несколько базовых операций по вводу/выводу файлов, чтобы сравнить производительность ZFS и `ext4`, используя утилиту `time` для измерения времени выполнения. Команда `time` измерит выполнение следующих двух примитивных операций:

```
$ dd if=/dev/urandom of=2GB bs=64M count=32 iflag=fullblock
$ cp 2GB aCору
```

Первая создает 2-ГБ файл, используя `dd`, а вторая создает копию 2-ГБ файла, созданного командой `dd`. Обе эти команды будут выполнены в конфигурации RAID 0 ZFS, конфигурации RAID-Z ZFS и в файловой системе `ext4`.

График вверху представляет реальное поле из вывода команды `time`. Как видно из графика, ZFS прекрасно справляется: она не уступает по скорости `ext4` и предлагает массу преимуществ. Однако операция `cp` в разделе `ext4` была намного быстрее, чем обе операции копирования в файловой системе ZFS.

Итак, стоит ли использовать ZFS? Мы бы сказали, что если ваше оборудование ей соответствует, то ответом определенно будет «да»! **LXF**

```
2. mtsouk@LTTng: ~ (ssh)
mtsouk@LTTng:~$ sudo zpool destroy LXF
mtsouk@LTTng:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            2.0G   0  2.0G   0% /dev
tmpfs           396M   6.1M 390M   2% /run
/dev/sda1       16G   13G  2.7G  83% /
tmpfs           2.0G  192K  2.0G   1% /dev/shm
tmpfs           5.0M   4.0K  5.0M   1% /run/lock
tmpfs           2.0G   0  2.0G   0% /sys/fs/cgroup
tmpfs           396M   32K  396M   1% /run/user/108
tmpfs           396M   0  396M   0% /run/user/1000
mtsouk@LTTng:~$ sudo zpool create -f LXF raidz /dev/sdb /dev/sdc /dev/sdd
mtsouk@LTTng:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            2.0G   0  2.0G   0% /dev
tmpfs           396M   6.1M 390M   2% /run
/dev/sda1       16G   13G  2.7G  83% /
tmpfs           2.0G  192K  2.0G   1% /dev/shm
tmpfs           5.0M   4.0K  5.0M   1% /run/lock
tmpfs           2.0G   0  2.0G   0% /sys/fs/cgroup
tmpfs           396M   32K  396M   1% /run/user/108
tmpfs           396M   0  396M   0% /run/user/1000
LXF            5.8G   0  5.8G   0% /LXF
mtsouk@LTTng:~$
```

► Можно создать раздел RAID-Z ZFS всего одной командой `zpool!` Недостатком будет меньше места на диске, чем в конфигурации RAID 0, на тех же жестких дисках.



При работе с операциями, затрагивающими конфигурацию ваших файловых систем, всегда стоит вначале сделать резервную копию системы, особенно при работе с производственными серверами. Однако наличие тестового компьютера для проверки ваших команд сделает вашу жизнь намного легче!

## Другие функции ZFS

Наш урок представил немалую часть функций ZFS и их команды администрирования, но ZFS умеет куда больше.

Прежде всего, для обеспечения целостности данных ZFS использует контрольные суммы. Можно применять один из трех алгоритмов контрольных сумм: `fletcher2`, `fletcher4` и `sha256`. Хотя контрольные суммы можно отключить, делать такое не рекомендуется, поскольку это практичная функция для защиты ваших данных.

Вдобавок ZFS поддерживает сжатие данных (по умолчанию отключено). Хотя сжатие может

снизить производительность файловой системы ZFS, бывают случаи, когда производительность чтения и записи, наоборот, повышается, потому что из-за сжатия производится доступ к меньшему количеству блоков для чтения или записи тех же самых данных. Но если у вас много места на диске, то сжатие данных вам, возможно, не понадобится.

ZFS также поддерживает `Copy on Write` — технологию, которая обеспечивает целостность данных на вашем диске. Проще говоря, данные при изменении не перезаписываются, как в большинстве файловых систем. Вместо этого новые данные

сохраняются на новых блоках диска, а затем изменяются указатели на данные.

ZFS также поддерживает `Snapshot Replication`, которая происходит на уровне блоков. Это означает, что синхронизировать нужно только разницу в блоках, что делает процесс синхронизации очень эффективным. Хотя, возможно, это не покажется значимым, на самом деле, это позволяет вам дублировать ваши данные в сети.

И наконец, ZFS поддерживает пулы данных, включающие комбинацию устройств SSD, SAS и SATA, не требуя дополнительных забот.

ЧАСТЬ 10

# IoT: Дом, который послушен нам

Максим Черепанов выдал полномочия по своему Послушному Дому всем членам семьи и задумался о последствиях.



Наш эксперт

**Максим Черепанов** — заместитель директора в небольшой коммерческой фирме. По образованию железнодорожник-управленец. Linux занимается с 2008 г. Чем дольше это делает, тем дальше от Windows. За эти годы разучился чистить реестр и забыл слово «антивирус».



Э то снова я, со своим Послушным Домом. Соскучились? Я тоже. За время, проведенное с вами, мы умудрились обеспечить наш ПД аж 6-ю каналами передачи и 4-мя каналами приема управляющих и информационных сигналов, обратив внимание на рис. 1.

Также мы соединили с ПД видекамеру и встроили изображение, передаваемое с нее, в web-интерфейс. На схеме показаны только два канала нагрузки (дублирующие я рисовать не стал); также опущена силовая часть.

Всё? Нет, отнюдь (эта фраза должна быть уже знакома). Успокаиваться я вам не дам. Давайте-ка поразмышляем о визуализации процессов, происходящих в ПД. Причем не системных, о которых мы говорили при создании файлов-маркеров (или флагов), а о значимых для пользователя.

Смоделирую для вас ситуацию: вы выдали всем своим домочадцам логины и пароли от системы управления ПД. Всё происходит прекрасно до того момента, когда произойдет конфликт интересов: вам надо включить торшер в спальне, поскольку вы читаете книгу. А старший сын, зайдя в дашборд с работы, увидел по индикации на web-панели, что горит свет. И выключает его. Ничего страшного не происходит, вы включите его снова. Но неплохо бы знать, кто помешал вам читать (не для того, чтобы снять ремешок и напомнить сыну золотые годы детства, а чтобы уведомить о том,

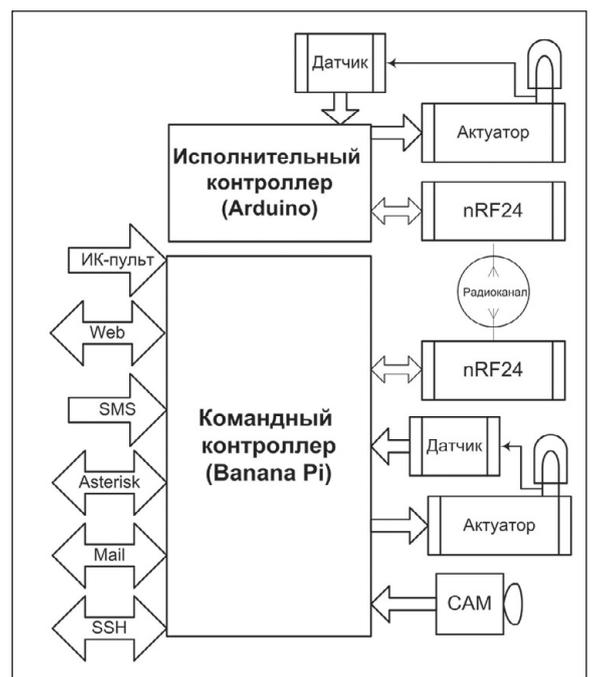
что вы читаете и свет горит не просто так). Как вы узнаете, кто это сделал? Разумеется, в любой мало-мальски приличной системе должен вестись подробный лог-файл всех действий. Этой темы мы еще не касались (если серия будет продолжаться, то коснемся и этого), и логи у нас не ведутся — для упрощения учебного проекта. Да и для того, чтобы посмотреть лог-файл, надо подключиться по SSH и просмотреть логи нескольких служб.

Так как же быть?

Есть предложение: для вывода небольшого количества текущей информации будем использовать LCD-дисплей. Разумеется, технология эта не то чтобы не нова, а вполне заслуживает грамоты «За заслуги в прошлом веке». Но не торопитесь захлопывать журнал — при помощи небольшого дисплейчика и 4 проводов мы с вами сможем сделать удивительно простую и эффективную систему наблюдения за процессами, происходящими в ПД.

## И снова немного теории

Помучаю вас экстрактом информации о самом устройстве и о способе его подключения.



» Рис. 1. Блок-схема ПД.

Итак, что у нас в руках? Невероятно популярный 15 лет назад и применяемый до сих пор повсеместно LCD-дисплейчик (присмотритесь к кофейным автоматам — на большинстве из них применяются самый популярный вариант: 2 ряда по 16 знакомест с синей подсветкой).

Разберемся с ним подробнее. Основа его микросхема HD44780 (KS0066) — контроллер монохромных жидкокристаллических знаковинтегрирующих дисплеев с параллельным 4- или 8-битным интерфейсом. Разработан фирмой Hitachi в 1980-е гг. Обычно контроллер и дисплей выполнены конструктивно на одной печатной плате и имеют 14–16 выводов для подключения к микроконтроллеру или микрокомпьютеру (2 дополнительных — это питание подсветки; вариант с 14-ю выводами обычно применяли в факсах и подсвечивали дисплей отдельно). Сразу обращаю ваше внимание, что подсветка потребляет достаточно большой ток, до 150 мА. Это регулируется токоограничивающим резистором — обычно хватает 100 Ом, ток при этом составит около 50 мА (напряжение 5 вольт). Сам дисплейчик потребляет мизерный ток, единицы миллиампер.

Распиновку остальных 14-ти выводов вы без труда найдете сами. Но суть не в этом, а в том, что эти 14 выводов просто некуда подключать к микрокомпьютеру в первоизданном виде — слишком много ножек.

Выручит нас другой миниатюрный девайс — платка сопряжения LCD-дисплея на HD44780 и шины I2C, которая требует всего 4 провода. Ну, и как у нас водится, немного про эту самую шину.

I2C (Inter-Integrated Circuit — примерно «схема внутренней связи»), у нас читается как «и-два-цэ» — последовательная шина, придуманная Philips еще в 80-х годах прошлого века (как много было придумано в «золотые 1980-е», верно?). Задумывалась как простая шина для связи блоков внутри устройства — и получилась очень удачной, поверьте (сегодня каждый производитель, выпускающий МК, предусматривает свой аппаратный I2C). Правда, название в Philips запатентовали, и до 2007 г. все придумывали своим вариантам собственные названия.

Производители законченных блоков и модулей зачастую используют именно I2C для общения с внешним миром (блоки телевизоров, магнитол, дисплеи, некоторые камеры в мобильных телефонах и т. п.), а микросхем периферии вообще на любой вкус: АЦП/ЦАП, памяти разнообразны, часы реального времени, расширители вводов-выводов, гироскопы, акселерометры, компасы, драйверы светодиодов и матриц, ШИМ-контроллеры, синтезаторы частот и вообще всё, что душе угодно.

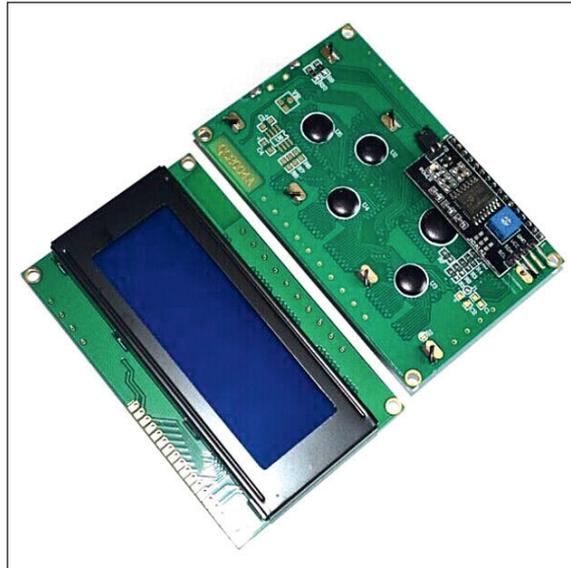
«Недостатков» у этой чудо-шины два: скорость работы и ограничение на число адресов в сети, классически это 100 кбит/с и 127 устройств (ну вы поняли, почему первое слово — в кавычках). Стандарт 1992 г. предусматривает уже скорости как 100 кбит/с (low-speed), так и 400 кбит/с (fast-speed) и уже 10-битную адресацию, т. е. максимум — 1203 адреса. А стандартом от 1998 скорость раскрутили аж до 3,4 Мбит/с (Hs-mode).

Мне лично за глаза и первого стандарта, да и мороки с этим Hs-mode значительно больше.

Адреса (из 127) обычно жестко «вшиты» в микросхему, но частично несколько бит адреса «выводят» на ножки — перемычками (на питание или землю) можно задать эти биты и подключить несколько одинаковых микросхем на одну шину.

Физически, шина I2C представляет собой два провода (не считая земли и питания), притянутые к плюсу резисторами 1–10к (это строго обязательное условие).

Но сильно глубоко вдаваться в подробности железа я вас не заставлю, потому что выручит нас от всей этой информации I2C-расширитель портов на базе чипа PCF8574. Описана эта микросхема достаточно подробно, но даже и это описание нам не понадобится!



► Рис. 2. LCD-дисплей 20×4.

Потому что есть возможность купить сразу готовое изделие, точно вставляющееся в 16 отверстий на плате с HD44780 и LCD-дисплеем (или даже сразу готовый LCD со впаянной платкой).

Вернемся ненадолго к дисплеям: существует много вариантов, как разных размеров (8×2, 16×2, 20×2, 20×4, 40×2, 40×4), так и разных цветов и подсветок. У меня был выбор: 16×2 или 20×4. Выбрал первый вариант: информации на них можно уместить прилично (при хорошей фантазии), а размер и ток потребления меньше. Приобрел сразу две шильды с PCF8574. Спаял оба комплекта, 20×4 (рис. 2) пока отложил, обязательно применю его для конечного варианта ПД.

Ну, а мы займемся «младшим братом», рис. 3.

И теперь главный бонус (или просто приятная новость): на гребенке нашего BananaPi уже предусмотрены специальные выводы именно для подключения шины I2C — это GPIO3 и GPIO5 (SDA и SCL). И схема подключения необыкновенно тривиальна (рис. 4).

Синие провода на схеме вам паять и соединять не надо, это собственно коннект платы расширения и платы LCD.

Владельцам RaspberryPi (как и Cubieboard) также можно радоваться, там эти же пины находятся на том же месте.

Теперь хватит железа, пора приступить к освоению софта.

## Немного дегтя

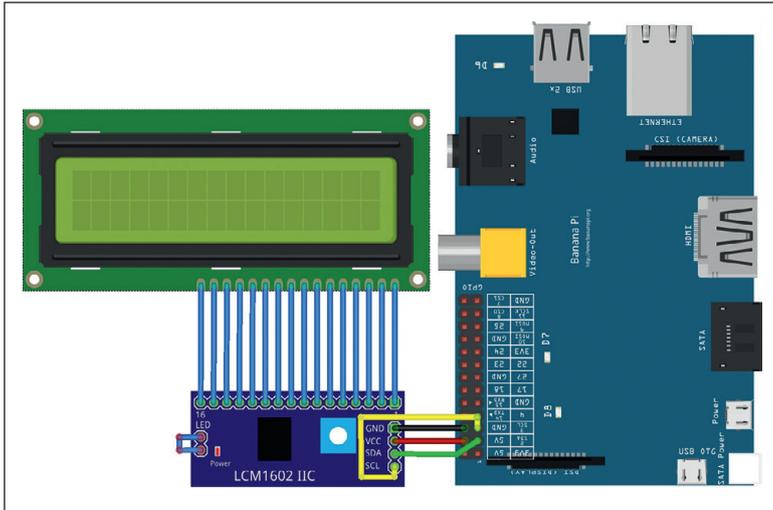
На первый взгляд, всё было превосходно. Я решил действовать в строгом соответствии с фэн-шумом: протестировать доступность

»



► Рис. 3. LCD-дисплей 16×2.

» Подпишитесь на печатную или электронную версии на [www.linuxformat.ru/subscribe!](http://www.linuxformat.ru/subscribe!)



➤ Рис. 4. Схема соединения LCD и BananaPi.

устройства, подобрать драйвер, набросать тестовый скрипт и с чистой совестью встраивать его в ПД.

### ➤ Этап первый: тест доступности

Для этого будем использовать готовый комплект *i2c-tools*, в его составе — 4 программы-утилиты: *i2cdetect*, *i2cdump*, *i2cget*, *i2cset*.

Устанавливаем:

```
# apt-get install i2c-tools
```

Кратко о частях (пригодится):

» *i2cdetect* Используется для проверки подключенных устройств — выводит адреса slave-устройств.

» *i2cdump* Программа для снятия дампа данных.

» *i2cget* Получаем значение нужного нам регистра в подключенном устройстве. Опрашивается в формате

```
i2cget <шина> <чип> <регистр>
```

» *i2cset* Записываем значение в нужный регистр подключенного устройства, команда в формате

```
i2cset <шина> <чип> <регистр> <значение>
```

Например: `# i2cset 0 0x2d 0x10 0x22`, то есть пишем значение `0x22` в регистр `0x10` устройства по адресу `0x2d` на шине `0`.

Прежде чем пользоваться этой утилитой, надо включить шину *i2c* в устройстве. Для этого снова придется включить ее в конфигурации, содержащейся в файле `/bin/script.bin`. В случае BananaPi это символическая ссылка на файл. Узнаем, на какой:

```
$ ls -l /boot/script.bin
```

```
lrwxrwxrwx 1 root root 16 Jun 22 2016 /boot/script.bin -> bin/bananapi.bin
```

Преобразуем файл `bananapi.bin` в читаемый формат с помощью пакета утилиты *bin2hex* (процесс я описывал в статье про шину *spi* и подключение радиомодуля). После преобразования открываем получившийся файл `bananapi.hex` и ищем раздел `[twi]` (так обычно называют «патентованную» шину *i2c* в сторонних контроллерах и микрокомпьютерах). Она находится в разделе `[twi2_para]`, и для включения шины его надо привести к следующему виду:

```
[twi2_para]
```

```
twi2_used = 1
```

```
twi2_scl = port:PB20<2><default><default><default>
```

```
twi2_sda = port:PB21<2><default><default><default>
```

Не пытайтесь найти соответствие этой записи и физическим точкам подключения, я ее не нашел. Но так написано в технической документации, и оно работает.

Приятная новость для тех, кто использует ОС от LeMaker: шина включена в сборке изначально. Для тех, кто использует сборки

Armbian, а также для владельцев Raspberry и Cubieboard: вам надо включать шину способом, указанным в ваших подсказках. Не сомневайтесь, она есть.

Ну что, пробуем? Помним, что шина у нас с индексом 2, подключаем дисплей по рисунку, обеспечиваем хорошее питание всего комплекса и проверяем:

```
# i2cdetect -y 2
```

```
0 1 2 3 4 5 6 7 8 9 a b c d e f
```

```
00: -----
```

```
10: -----
```

```
20: -----
```

```
30: ----- 3f
```

```
40: -----
```

```
50: -----
```

```
60: -----
```

```
70: -----
```

Итак, наш дисплейчик доступен и имеет адрес `3F` на второй шине, первый этап пройден.

### ➤ Этап второй: подбор драйвера

Вот тут и появились те грабли, или ложка дегтя. Оказалось, что дисплейчик не совместим со стандартными `lcd4linux` и `lcdproc` (это наиболее популярные драйвера для подключения LCD к микрокомпьютерам). Дело в распойке. Я пытался многими способами подружить эти драйвера и свой девайс, но получались либо квадраты, либо кракозябры. Иногда экран гаснул, потом опять загорался (вот так я и узнал, что подсветку можно включать и выключать программно). Но полезная информация не появлялась. Как оказалось, это достаточно частая история для девайсов, произведенных в Поднебесной и продаваемых на AliExpress. К счастью, на большинство граблей в этом мире мы наступаем не первыми. Поиск в русскоязычном Интернете дал положительный результат — некто Владимир Романов эту проблему уже решил, написав драйвер именно для таких случаев: [https://github.com/vvromanov/cb\\_i2c\\_lcd](https://github.com/vvromanov/cb_i2c_lcd).

Что мы можем получить по этой ссылке? Готовый драйвер, стоит только его допилить для своих нужд и скомпилировать. Файл `i2c-dev.h` — это заголовочный файл, его трогать не надо. Файл `i2c_lcd.c` — это собственно сам драйвер. Вот его мы немного и изменим для своих нужд.

Для начала внесем в него данные о своей шине и адрес устройства (удивлен, что автор скрипта не сделал это через константы или переменные; драйвер-то классный, но заточен под совершенно конкретное подключение).

Открываем файл `i2c_lcd.c` в текстовом редакторе и меняем строку, где задаются эти параметры:

```
int fd=init_i2c_bus(2, 0x3f);
```

`2` — это номер шины, `0x3F` — адрес дисплея.

В принципе с этого момента он работает и так. Только он заточен под формат `20x4` (если у вас такой дисплей, то половина дела сделана за вас В. Романовым). А для `16x2` надо сделать следующее.

Отыщите функцию `void write_line(int line, const char* s)`, это запись символов в дисплей в зависимости от линии. Удалите `case 2` и `case 3`. Далее, везде в этой функции, где упоминается переменная `len` (3 раза), замените число `20` на `16` — это количество знакомест. Если их у вас `8`, то замените на `8`. Если оставить большее значение, страшного ничего не будет: драйвер просто будет писать символы в пустоту. Но это же не комильфо, верно?

Потом надо удалить в конце файла следующий блок:

```
}
if (argc>=4) {
```

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

```
write_line(2,argv[3]);
}
```

А дальше мой личный тюнинг: для эстетов или экономистов (я к вопросу о питании, это почти самая насущная тема для микрокомпьютеров). Подсветка нагло потребляет огромное количество энергии, да и светит ярко (особенно ночью). Стоило бы ее отключать, да вот только в драйвере про это ни слова.

Впрочем, намек на это есть. В начале файла можно найти такую строку:

```
#define LCD_BL 1<<3
И задается переменная:
uint32_t lcd_backlight=LCD_BL;
```

Пин 3 в данном варианте LCD отвечает за включение и отключение подсветки. Да вот беда, это никак не используется (или я не разобрался, как). Исправим так, чтобы всё работало и было понятно?

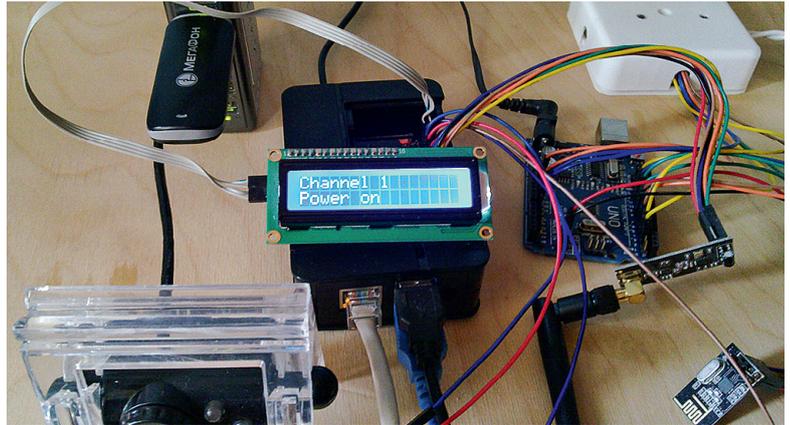
Ищем функцию входа в функцию `int main`, и меняем ее. Привожу функцию полностью, во избежание ошибок:

```
int main(int argc, char* argv[])
{
int fd=init_i2c_bus(2, 0x3f);
if (argc==1) {
write_pulse(LCD_D5 | LCD_D4);
usleep(5000);
pulse();
usleep(200);
pulse();
usleep(200);
write_pulse(LCD_D5);
usleep(5000);
write_cmd(0x28);
write_cmd(0x06);
write_cmd(0x0C);
} else {
if (argc>=2) {
if ((strcmp(argv[1], "b_light_off")) & (strcmp(argv[1], "b_light_on"))!=0) {
write_line(0,argv[1]);
} else {
if (strcmp(argv[1], "b_light_off")==0) {
lcd_backlight=0;
i2c_write_byte(lcd_backlight);
}
if (strcmp(argv[1], "b_light_on")==0) {
lcd_backlight=1<<3;
i2c_write_byte(lcd_backlight);
}
}
}
if (argc>=3) {
write_line(1,argv[2]);
}
}
close(fd);
```

Теперь мы сможем отключать или включать подсветку, передавая драйверу параметры `b_light_on` или `b_light_off` в качестве первого аргумента. Недостаток этого способа типа «решение в лоб» таков, что мы не сможем вывести такие текстовые данные (`b_light_on` или `b_light_off`) в первую строку. Но вряд ли это так уж критично.

Теперь компилируем; для этого автор скрипта любезно предоставил нам `CmakeLists.txt`. Как пользоваться `cmake`, я не буду писать, это просто и быстро, и есть в Интернете.

После этого мы имеем готовый двоичный драйвер `i2c_lcd`. Переложите его в любую удобную папку — и можете запускать,



► Рис. 5. Индикация включения на LCD-дисплее.

указывая первым аргументом то, что надо записать в первую строку, вторым — во вторую, например:

```
./i2c_lcd "line1" "line2"
Разумеется, кроме тех случаев, когда надо выключить или включить подсветку:
./i2c_lcd b_light_off
./i2c_lcd b_light_on
```

Для примера я покажу кусочек своего диалплана (начало) для Asterisk'a, а вы сможете сравнить его с тем, что был опубликован ранее:

```
[from-gsm]; From GSM
exten => s,1,Answer
exten => s,n,Playback(hello)
exten => s,n,Set(CHANNEL(language)=ru)
exten => s,n,SayUnixTime(.,\time\kMQdb)
exten => s,n,Set(RESULT=${SHELL(/home/user/i2c_lcd b_light_on)})
exten => s,n,Set(RESULT=${SHELL(/home/user/i2c_lcd "Called:" "${CALLERID(num)}"))})
exten => s,n,Wait(1)
exten => s,n,Authenticate(1887)
exten => s,n,Gotof("${CALLERID(num)}" = "+79036666666"?999,1;)
exten => s,n,Gotof("${CALLERID(num)}" = "+79607777777"?999,1;)
exten => s,n,Set(RESULT=${SHELL(/home/user/i2c_lcd "Asterisk" "hangup")})
exten => s,n,Wait(1)
exten => s,n,Set(RESULT=${SHELL(/home/user/i2c_lcd "Wait" "")})
exten => s,n,Set(RESULT=${SHELL(/home/user/i2c_lcd b_light_off)})
exten => s,n,Hangup()
```

Принимается звонок, проговариваются приветствие и текущие время/дата. Далее вступает в действие наш драйвер: включаем подсветку, выводим на экран информацию о номере звонящего. Ждем 1 секунду и просим ввести код аутентификации. После ее прохождения проверяем, можем ли мы дать управление (проверяются 2 номера). Это 2-уровневая система безопасности. Если всё в порядке, то управление передается на контекст 999. Если нет — пишем на экран Asterisk hangup, потом Wait, гасим экран и кладем трубку. И это всё у меня работает!

Я не привожу весь диалплан, он довольно большой. Его версию без управления дисплеем вы уже видели. Приведу фото момента (рис. 5), когда на экран выводится информация о включении 1 канала нагрузки.

Собственно говоря, на сегодня всё. Теперь вы можете постоянно видеть, что творится внутри вашего ПД — это ведь совсем нетрудно (и недорого), правда?

До новой встречи! LXF

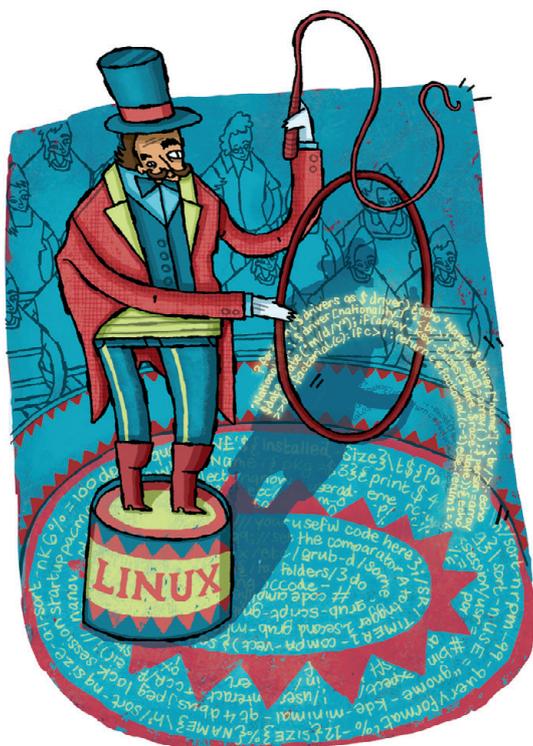
# Загадка LXF: Расчет числа π

Михалис Цукалос научит вас вычислять число π с желательной точностью с помощью нежелательных математических навыков.



**Наш эксперт**

**Михалис Цукалос** — администратор UNIX, программист-математик и администратор баз данных, который любит писать статьи и осваивать что-то новое.



```

1 \documentclass[12pt, a4paper, landscape]{article}
2
3 \usepackage{amssymb, amsmath}
4 \usepackage[left=1.5in, right=1in, top=1.75in, bottom=2in]{geometry}
5
6 \begin{document}
7
8 \Large
9
10 \[
11 \pi = \frac{1}{2^6} \sum_{n=0}^{\infty} \frac{\left( (-1)^n \right)}{2^{10n}}
12 \left( -\frac{2^5}{4n+1} - \frac{1}{4n+3} + \frac{2^8}{10n+1} - \frac{2^6}{10n+3} - \frac{2^2}{10n+5} - \frac{2^2}{10n+7} + \frac{1}{10n+9} \right)
13 \]
14
15 \[
16 \frac{1}{\pi} = 12 \sum_{k=0}^{\infty} \frac{-1^k (6k)! (13591409 + 545140134k)}{(3k)! (k!)^3 640320^{3k + \frac{3}{2}}}
17 \]
18
19 \end{document}

```

▶ Вот код LaTeX, используемый для создания формулы ниже. Эта формула хранится в файле formula.tex. LaTeX — мастер записи математических формул!

понятиях в языках программирования Go, Perl, Python 3 и C, а также научитесь работать со сложными уравнениями в LaTeX. Нас ждут веселые времена! [Ред.: — Ух ты!]

Существует множество методов и алгоритмов, позволяющих рассчитывать число π с требуемой точностью. Преимущество самостоятельного вычисления π заключается в том, что так вы лучше контролируете процесс, а значит, сможете вносить небольшие изменения в код или алгоритм независимо от результатов работы других пользователей и не рассматривая весь процесс как черный ящик, который просто возвращает числа.

## Выбор из двух...

Вообще говоря, при решении численных задач существует два основных подхода. При первом вы пытаетесь сохранить результат, чтобы использовать его в других вычислениях, а при втором вам

## Примите участие!

Посоревнуйтесь с Михалисом! Отправьте свой код по адресу электронной почты **LXF: linuxformat@futurenet.com**, с темой: **Загадка LXF227**. Убедитесь, что ваш код компилируется без ошибок (также не забудьте хорошо его задокументировать код и снабдить подробными комментариями), и вы получите шанс обойти нашего умника.

**Н**а этом уроке мы рассчитаем число π без использования библиотеки или модуля. Хотя эта задача выглядит легкой, она может показаться интереснее, когда вы поймете, что число π нужно вычислить с заданной точностью. А стало быть, простого вычисления недостаточно: вам понадобится метод, который является и быстрым, и точным.

Настоящая польза от решения задач в программировании заключается в том, что попутно вы накапливаете интересные знания и опыт. Так, например, из этого урока вы узнаете о некоторых

$$\pi = \frac{1}{2^6} \sum_{n=0}^{\infty} \frac{(-1)^n}{2^{10n}} \left( -\frac{2^5}{4n+1} - \frac{1}{4n+3} + \frac{2^8}{10n+1} - \frac{2^6}{10n+3} - \frac{2^2}{10n+5} - \frac{2^2}{10n+7} + \frac{1}{10n+9} \right)$$

$$\frac{1}{\pi} = 12 \sum_{k=0}^{\infty} \frac{-1^k (6k)! (13591409 + 545140134k)}{(3k)! (k!)^3 640320^{3k + \frac{3}{2}}}$$

▶ Для расчета числа π с требуемой точностью применяются формула Бельяра и алгоритм Чудновского.

## О расчете числа $\pi$

Итак, задача нашего урока — расчет числа  $\pi$  с произвольным количеством знаков. Это означает, что в выбранном нами методе вычисления не должно быть ограничения на количество знаков после запятой.

Одна вещь, которую обязательно нужно учитывать — это выбор языка программирования, который поддерживает числа с плавающей запятой с большим количеством цифр. Это вызвано тем, что вам, возможно, придется рассчитать число  $\pi$

с 1000 или даже 1 000 000 цифр после запятой, что не так уж просто. Например, C не поддерживает очень большие числа с плавающей запятой (если вы не хотите искать внешнюю библиотеку, в которой есть специальный тип данных, подходящий для решения этой задачи).

Список методов вычисления числа  $\pi$  с требуемой точностью включает формулу Бэйли–Боружайна–Плуффа [Bailey–Borwein–Plouffe], формулу Бельяра [Bellard], алгоритм Чудновского, алгоритм

Гаусса–Лежандра [Gauss–Legendre], формулу Лейбница [Leibniz], алгоритм Лю Хуэя [Liu Hui], алгоритм Боружайна, формулу Виета [Viète], произведение Уоллиса [Wallis], метод быстрого вычисления E-функций и алгоритм Чжао Юцина [Zhao Youqin]. Большинство из них вычисляют сумму последовательности чисел — чем больше взята чисел, тем точнее результат.

Как видите, концепция расчета числа  $\pi$  интересна многим!

просто надо распечатать результат. Для числа  $\pi$  второй подход означает, что программе просто надо будет печатать цифру за цифрой, не выполняя каких-либо дополнительных действий с конечным результатом.

Первый подход сложнее, так как вам надо найти способ хранения чисел с плавающей запятой с произвольным количеством знаков. Но по сути вам незачем хранить произвольное количество знаков — достаточно хранить очень много цифр!

Так или иначе, довольно разговоров — пора начать знакомиться с некоторыми вещами, которые помогут нам решить эту задачу!

### Два способа расчета

Здесь мы попытаемся демистифицировать два способа расчета числа  $\pi$ , чтобы лучше понять, как они работают и что означают некоторые математические символы. Тогда вы сообразите, как преобразовать математическую формулу в форму, пригодную для языка программирования.

На рис. внизу стр. 84 показаны две формулы для расчета числа  $\pi$ . Верхняя формула — формула Бельяра, нижняя используется алгоритмом Чудновского. Хотя обе формулы поначалу выглядят сложными, на самом деле они относительно просты.

Первое, что можно заметить — формула Бельяра выглядит сложнее формулы Чудновского; а второе — что формула Чудновского вычисляет значение, обратное значению  $\pi$ . Однако по обратному значению  $\pi$  легко определить и само число  $\pi$ . Большой греческий символ  $\Sigma$ , используемый в обеих формулах, означает, что необходимо сложить результаты всех итераций.

Вдобавок все слагаемые в алгоритме Чудновского больше нуля для всех четных значений  $k$  и меньше нуля для всех нечетных значений  $k$ . Это означает, что при приближении к числу  $\pi$  сумма всех значений то возрастает, то уменьшается. То же касается и формулы Бельяра. Это означает, что оба способа пытаются уточнить значение  $\pi$ , добавляя и вычитая некие величины из ранее вычисленной суммы.

С другой стороны, расчеты для формулы Бельяра проще, поскольку нам приходится иметь дело с меньшими величинами. Чтобы лучше понять это, примем во внимание тот факт, что для  $k=5$  факториал  $3k$  равен 1 307 674 368 000 — представьте, как огромно это число для действительно больших значений  $k$ ! Кроме всего этого, обе формулы должны вычислять некоторые дроби. Это легко при условии, что вы можете хранить результаты как переменные с плавающей запятой с большим числом знаков.

На рис. сверху стр. 84 показан документ LaTeX, используемый для создания двух формул. Имя файла с кодом LaTeX — `formulas.tex`.

По прочтении этого раздела важно осознавать, что как только вы вполне поймете формулу, которую намерены использовать,

разработка программы расчета числа  $\pi$  по этой формуле, на любом языке программирования, станет сравнительно простой задачей.

### $\pi$ на Python 3(14)

В этом разделе будет представлена первая версия программы на Python, которая рассчитывает число  $\pi$  по формуле Бельяра. Всё, что надо указать программе — это требуемое количество знаков после запятой.

Программа называется `calcPi.py`, а самый важный код на Python 3 — это определение функции `Pi()`, которая делает всю трудную работу:

```
def Pi():
    pi = Decimal(0)
    k = 0
    while k < numberOfDigits:
        pi += (Decimal(-1)**k/(1024**k))*(Decimal(256)/(10**k+1) +
        Decimal(1)/(10**k+9) - Decimal(64)/(10**k+3) - Decimal(32)/(4**k+1) -
        Decimal(4)/(10**k+5) - Decimal(4)/(10**k+7) - Decimal(1)/(4**k+3))
        k += 1
```

### Скорая помощь

Простейший способ записать математическую формулу — воспользоваться TeX или LaTeX. Но можно также разработать формулу с помощью растровых или векторных инструментов.

```
6
7 printAll = False
8 if len(sys.argv) == 2:
9     numberOfDigits = int(sys.argv[1])
10 elif len(sys.argv) == 3:
11     numberOfDigits = int(sys.argv[1])
12     if sys.argv[2] == "-print":
13         printAll = True
14 else:
15     print("Not enough arguments!")
16     sys.exit(0)
17
18 getcontext().prec=numberOfDigits
19
20 def Pi(n):
21     t = Decimal(0)
22     pi = Decimal(0)
23     deno = Decimal(0)
24     k = 0
25     n = int(n / 20)
26     for k in range(n):
27         t = ((-1)**k)*(factorial(6*k))*(13591409+545140134*k)
28         denominator = factorial(3*k)*(factorial(k)**3)*(640320**(3*k))
29         pi += Decimal(t)/Decimal(denominator)
30         if printAll == True:
31             print(pi)
32     pi = pi * Decimal(12)/Decimal(640320**Decimal(1.5))
33     pi = 1/pi
34     return pi
35
36 print(Pi(numberOfDigits))
37
```

» Реализация на Python 3 алгоритма Чудновского, который используется для вычисления числа  $\pi$  с требуемой точностью. Код чист и легко читается.

» Подпишитесь на печатную или электронную версии на [www.linuxformat.ru/subscribe!](http://www.linuxformat.ru/subscribe!)

```
pi = pi * 1/(2**6)
return pi
```

Самое главное здесь в том, что функция `Pi()` возвращает  $\pi$  как число, которое можно сохранить в других переменных и использовать в других расчетах.

Также важен следующий вызов, поскольку он позволяет определить желаемую точность всех чисел с плавающей запятой, используемых в программе:

```
getcontext().prec = numberOfDigits
```

Запуск файла `calcPi.py` сгенерирует следующий вывод — нечто подобное и ожидается от программы расчета числа  $\pi$ :

```
$. /calcPi.py 120
```

```
3.141592653589793238462643383279502884197169399375105
82097494459230781640628620899862803482534211706798214
808651328230673
```

```
$. /calcPi.py
```

```
Не хватает аргументов!
```

Дополнительную информацию о модуле `decimal`, который используется в вычислениях, можно найти на странице документации <https://docs.python.org/3/library/decimal.html>.

## $\pi$ по алгоритму Чудновского

В коде Python 3 в этом разделе для расчета числа  $\pi$  применяется алгоритм Чудновского. Новая версия файла `calcPi.py` будет называться `Chudnovsky.py`.

При запуске файла `Chudnovsky.py` мы получим следующий результат:

```
$. /Chudnovsky.py 10
```

```
3.141592653
```

```
$. /Chudnovsky.py 120
```

```
3.141592653589793238462643383279502884197169399375105
```

```
82097494459230781640628620899862803482534211706798214
```

```
808651328230664
```

```
$. /Chudnovsky.py
```

```
Не хватает аргументов!
```

Кроме того, при запуске `Chudnovsky.py` с флагом `-print` результат выведется в более аналитическом виде, и вы сможете лучше понять весь процесс расчета и числа, формируемые на отдельных этапах вычисления:

```
$. /Chudnovsky.py 5 -print
```

```
1.3591E+7
```

```
3.1417
```

Рис. на стр. 85 показывает код на Python 3 файла `Chudnovsky.py`. Этот код достаточно прост — при условии, что вы поняли формулу Чудновского.

## С сложен прост!

Теперь достанем оружие помощнее и рассчитаем  $\pi$  на языке C. Но на сей раз воспользуемся другим подходом. Вместо формулы мы возьмем дробь, очень близкую к значению  $\pi$  — до определенного количества знаков после запятой. Получив дробь, мы будем вручную делить числитель на знаменатель, а затем остаток на знаменатель, до тех пор, пока не достигнем желаемого количества знаков числа  $\pi$ .

Итак, в этом разделе мы воспользуемся дробью  $3126535/995207$ , но при желании вы можете использовать свои собственные дроби, если они позволяют получить число  $\pi$  с хорошей точностью.

```
1  #!/usr/bin/perl -W
2
3  use strict;
4  use warnings;
5  use Math::BigFloat;
6
7
8  sub Pi {
9      my $Digits = shift;
10     $Digits = $Digits / 20;
11     my $pi = Math::BigFloat->new(0);
12     my $temp = Math::BigFloat->new(0);
13     my $k = Math::BigFloat->new(0);
14     my $k1k2k3 = Math::BigFloat->new(0);
15     my $k4k5k6 = Math::BigFloat->new(0);
16     while ($k < $Digits) {
17         $k1k2k3 = 1/(10*$k+9) - 64/(10*$k+3) - 32/(4*$k+1);
18         $k4k5k6 = - 4/(10*$k+5) - 4/(10*$k+7) - 1/(4*$k+3);
19         $temp = $k1k2k3 + $k4k5k6;
20         $pi += ((-1)**$k/(1024**$k))* (256/(10*$k+1) + $temp);
21         $k++;
22     }
23     $pi = $pi * 1/(2**6);
24     return $pi;
25 }
26
27 my ($numberOfDigits) = @ARGV;
28
29 if (not defined $numberOfDigits) {
30     die "Need one numeric argument!\n";
31 }
32
33 Math::BigFloat->precision(- $numberOfDigits );
34
35 my $PI = Math::BigFloat->new(0);
36 $PI = Pi($numberOfDigits);
37 print $PI . "\n";
38
```

➤ Вот код скрипта `calcPi.py` (на Perl), позволяющего рассчитать число  $\pi$ . Как и ожидалось, программа на Perl вышла относительно простой.

## Что считается быстрым?

Если вы собираетесь использовать число  $\pi$  в программе или если в вашей программе необходим расчет числа  $\pi$  с большим количеством знаков после запятой, то важнейшее значение имеет выбор самого скоростного метода. Однако для вычислений с невысокой степенью точности подойдут и медленные методы. Основная причина, по которой приходится отказаться от конкрет-

ного метода или алгоритма, заключается в том, что они не дают желаемой точности, а заменить отсутствие точности нельзя ничем. Кроме того, вам не подойдет метод, который возвращает  $\pi$  в виде строки, если  $\pi$  вам требуется в виде значения для дальнейших расчетов.

Наконец, если для вас главное — скорость вычислений, то сохранить число  $\pi$  в файл и читать его

из файла при необходимости — хороший обходной вариант, который сэконоит время на выполнение. В этом случае сгодится любой язык программирования, если он способен обеспечить необходимую вам точность.

Общий совет здесь заключается в том, чтобы выбирать метод, с которым не придется долго ждать результатов!

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

Как вы, несомненно, понимаете, чем больше значения числителя и знаменателя, тем выше будет конечная точность рассчитанного числа  $\pi$ .

Эта программа находится в файле **calcPi.c**, и она принимает в качестве аргументов командной строки числитель, знаменатель и количество знаков числа  $\pi$ , которые необходимо вычислить. В файле **calcPi.c** наиболее важен следующий код:

```
while(digits > 0) {
    printf("%d", num/den);
    num = (num % den) * 10;
    digits--;
}
```

Поскольку целочисленное деление несовершенно, цикл **while()** никогда не прекратится, то есть позволит получить приблизительное значение числа  $\pi$  с требуемым количеством знаков после запятой. Однако если вам нужна более высокая точность, этого метода следует избегать!

После компиляции и запуска файла **calcPi.c** вы получите следующий результат:

```
$. ./calcPi
calcPi numerator denominator digits
$. ./calcPi 312653 995207 10
Цифры неправильные!
$. ./calcPi 3126535 995207 10
3.1415926535
```

Как вы видите по способу получения результата, программа просто выводит цифры по мере их поступления, не сохраняя их. А значит, если вы не внесете изменения в программу, значение  $\pi$  будет потеряно.

Этот метод довольно быстр, но менее точен, чем в других программах, построенных по более сложным алгоритмам. Тем не менее, одним из преимуществ этой программы является то, что теперь вы научились обрабатывать аргументы командной строки на языке C!

## $\pi$ на Perl

Perl — эффективный и удобный язык программирования, позволяющий рассчитать число  $\pi$  по требуемому алгоритму. Программа на Perl будет вычислять  $\pi$  по формуле Бельера. Perl обрабатывает числа с плавающей запятой с высокой степенью точности. Для этого используется пакет **Math::BigFloat**.

Вы можете определить требуемую точность (до 500 цифр) и объявить новую переменную с такой точностью следующим образом:

```
Math::BigFloat->precision(-500);
my $pi = Math::BigFloat->new(0);
```

Запуск файла **calcPi.pl** даст следующий результат:

```
$. ./calcPi.pl 100
3,1415926535897932384626433832795028841971693993751058
209749445923078164062862089986280348253421170680
```

На рис. стр. 86 показан код Perl для файла **calcPi.pl**. Как видите, эта реализация программы похожа на реализацию в скрипте **calcPi.py**.

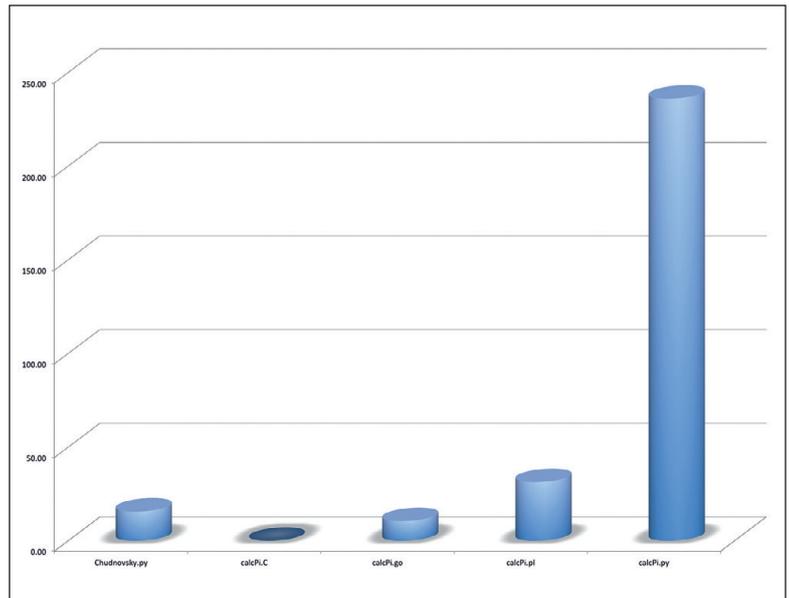
## $\pi$ на Go

Go — это современный язык программирования, предназначенный для точных математических расчетов. Программа на Go тоже будет вычислять  $\pi$  по формуле Бельера.

Секрет здесь состоит в использовании стандартного пакета Go — **math/big**, для более точных расчетов. Дополнительную информацию о пакете **math/big** можно найти на странице <https://golang.org/pkg/math/big>.

Назовем нашу программу **calcPi.go**. Два важных оператора Go в программе **calcPi.go** таковы:

```
temp := new(big.Float).SetPrec(precision).SetFloat64(0)
```



► Графическое представление поля **real** из вывода утилиты **time**. Каждая программа использовалась для вычисления 10 000 знаков числа  $\pi$ .

```
k7temp := new(big.Int).Exp(big.NewInt(-1), big.NewInt(int64(k)), nil)
```

Первый оператор создает новую длинную переменную с плавающей запятой (**big.Float**), а второй оператор вычисляет степень целого числа и сохраняет результат как переменную **big.Int**. Без переменных **big.Float** и **big.Int** вы не сможете вычислить  $\pi$  с желаемой степенью точности.

При запуске файла **calcPi.go** мы получим следующий результат:

```
$. go run calcPi.go
Предоставьте один числовой аргумент!
exit status 1
$. go run calcPi.go 100
3.1415926535897932569603993617387624040191831562485732
43493179283571046450248913467118511785
```

Осмотрев код Go в файле **calcPi.go**, вы заметите, что Go не слишком хорош в обработке больших чисел. Кроме того, код на Go является самым некрасивым из всех кодов нашего урока, а это значит, что в нем легче всего допустить ошибку. Тем не менее, программа **calcPi.go** свою работу делает.

## Оценка производительности

Пришло время [Ред.: — Очень забавно] для каждой программы расчета очень большого количества знаков числа  $\pi$  оценить производительность с помощью утилиты **time**. Хотя эта утилита не отличается особенной точностью, она даст вам хорошее представление о том, какой метод работает быстрее, а какой медленнее, и насколько эффективны используемые формулы.

На графике (выше) показано поле **real** из вывода команды **time**. Как видно из вывода, все программы были относительно медленными, за исключением программы на C — оно и понятно, ведь программа на C имеет наихудшую точность из всех. Программа Go идет следующей за **calcPi.c** по скорости и обладает большей точностью. Обратите внимание, что файл **calcPi.go** был скомпилирован с использованием команды **build calcPi.go**. Как и ожидалось, программы **Chudnovsky.py** и **calcPi.pl** медленнее **calcPi.c** и **calcPi.go**. Ну, а программа **calcPi.py** оказалась самой медленной из всех.

Можете ли вы придумать лучшую методику расчета числа  $\pi$ ? Вы реализовали этот метод на экзотическом языке программирования? Сообщите нам, и, возможно, вы найдете свой код в ближайшем номере **Linux Format!** LXF

## Скорая помощь

Почти все языки программирования поддерживают массивы или аналогичные структуры данных. Это означает, что для хранения полученных знаков числа  $\pi$  можно использовать большой массив. Массивы идеально подходят для этой задачи, поскольку они могут иметь огромное количество элементов.

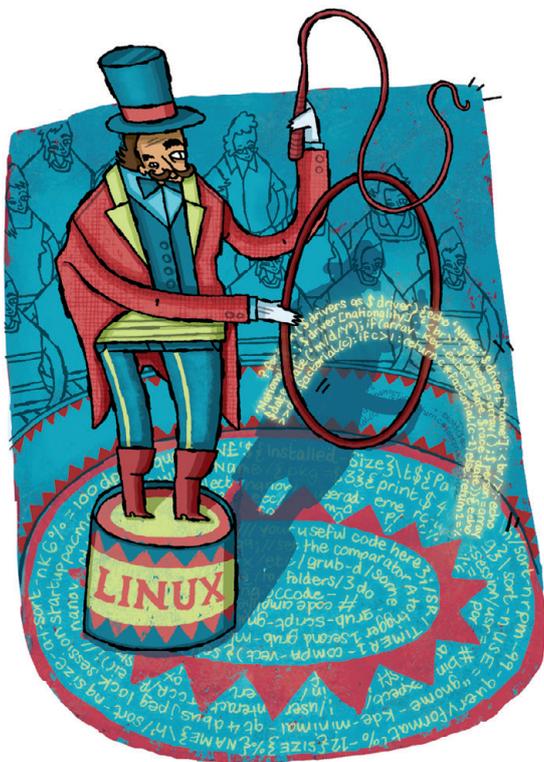
# Копирование: Ваша личная утилита `cp`

Вдохновившись пересмешником, **Михалис Цукалос** пишет облегченную версию `cp` на Python 3, чтоб безопасно копировать файлы.



**Наш эксперт**

**Михалис Цукалос** — администратор UNIX, программист, администратор баз данных и математик, который любит писать статьи и осваивать что-то новое.



```
$ cp /bin/cp /tmp/
$ ls -l cp
-rwxr-xr-x 1 mtsouk mtsouk 150824 Oct 19 22:09 cp
$ cp -n /bin/cp /tmp
$ cp -i /bin/cp /tmp
cp: overwrite '/tmp/cp'? y
```

Первая команда копирует исполняемый файл `cp` в текущий каталог. Как вы можете видеть, новый файл принадлежит текущему пользователю, а исходный файл — пользователю `root`. Последняя команда `cp` иллюстрирует использование ключа `-i`. Дополнительную информацию об этом и других ключах можно найти в выводе команды `man cp (1)`.

## Опознавание файлов и каталогов

Вам потребуется найти способ отличать каталоги от файлов. Именно это происходит в программе `fileORdir.py`:

```
#!/usr/bin/env python3
import os
import sys
if len(sys.argv) >= 2:
    what = str(sys.argv[1])
else:
    print('Не хватает аргументов!')
    sys.exit(0)
if os.path.isdir(what):
    print(what, '- это каталог!')
elif os.path.isfile(what):
    print(what, '- это файл!')
```

Метод `os.path.isdir()` определяет, является ли переданный ему параметр каталогом, а метод `os.path.isfile()` сообщает, является ли его параметр обычным файлом. Здесь следует отметить, что хотя использование сразу обеих проверок выглядит излишним, следует делать именно так, поскольку Linux поддерживает кучу других типов файлов, включая сокеты, ссылки и каналы.

При запуске файла `fileORdir.py` у нас получится следующий результат:

```
$. /fileORdir.py .
.- это каталог!
$. /fileORdir.py /usr
/usr - это каталог!
$. /fileORdir.py /bin/cp
/bin/cp - это файл!
```

Эта операция важна, поскольку если второй аргумент `cp` является каталогом, то это означает, что вы хотите скопировать в этот каталог исходный файл, указанный как первый аргумент

## Скорая помощь

Системное программирование — особая область программирования. Когда UNIX только появился, системные программы можно было писать только на C. Сейчас это можно делать и на других языках программирования, включая Python, Perl, Rust и Go.

**Н**а этом уроке я покажу, как реализовать утилиту командной строки `cp` на Python 3. Наша реализация на Python будет называться `руср.py`. Во всех операционных системах есть как минимум один способ копирования файлов, и Linux — не исключение. Чтобы понять, насколько важна команда `cp`, вы должны знать, что она находится в каталоге `/bin`. Этот каталог доступен, даже если загрузить компьютер Linux в однопользовательском режиме.

## Использование `cp`

В следующем выводе показаны различные варианты использования утилиты `cp`:

```
$ ls -l /bin/cp
-rwxr-xr-x 1 root root 150824 Mar 14 2015 /bin/cp
$ cp /bin/cp .
$ ls -l cp
-rwxr-xr-x 1 mtsouk mtsouk 150824 Oct 19 22:09 cp
$ rm cp
```

командной строки. А если первый аргумент является каталогом, то скрипт Python работать не будет.

## Работа с файлами

В этой части урока будут представлены методы Python 3, связанные с файлами. Эти методы позволят вам добавить некоторые дополнительные функции в **руср.py**. В следующем коде Python 3, который сохранен в файле **fileOperations.py**, показаны некоторые из методов для просмотра метаданных файлов:

```
#!/usr/bin/env python3
import sys
import os
import stat
from stat import *
if len(sys.argv) >= 2:
    filename = str(sys.argv[1])
else:
    print('Не хватает аргументов!')
    sys.exit(0)
fileMetadata = [(filename, os.lstat(filename))]
for name, meta in fileMetadata:
    if S_ISDIR(meta.st_mode):
        print('Это каталог')
    else:
        print('Это обычный файл')
        print(name, ' содержит', meta.st_size, ' байт')
print('File permissions:', stat.filemode(os.stat(filename).st_mode))
```

Разница между методами `os.lstat()` и `os.stat()` на Python 3 заключается в том, что `os.lstat()` не раскрывает символические ссылки, благодаря чему более безопасен по сравнению с `os.stat()`. При запуске файла **fileOperations.py** мы получим следующий результат:

```
$ ./fileOperations.py fileOperations.py
Это обычный файл
fileOperations.py содержит 502 байт
File permissions: -rwxr-xr-x
$ ./fileOperations.py .
Это каталог!
. содержит 4096 байт
File permissions: drwxr-xr-x
```

Как вы видите в разделе **File permissions** [Разрешения файла], для обычных файлов разрешения начинаются с дефиса, а для каталогов — с буквы `d` [directory].

## Так нечестно!

Python позволяет использовать в скриптах внешние команды. Для этого используется метод `os.system()`. Таким образом, вы можете реализовать копирование файлов с помощью самой утилиты

`ср`. Этот метод иллюстрируется следующим кодом, сохраненным в файле **cheating.py**:

```
#!/usr/bin/env python3
import os
import sys
if len(sys.argv) >= 3:
    source = str(sys.argv[1])
    destination = str(sys.argv[2])
else:
    print('Не хватает аргументов!')
    sys.exit(0)
os.system('cp '+ source + ' ' + destination)
```

Тем не менее, такой подход не считается правильным, и его следует избегать.

## Копирование файлов в Python 3

Большинство языков программирования, среди которых — C, Perl и C++, предлагают несколько способов копирования файлов. Python 3 — не исключение.

В принципе, если существует несколько способов решения задачи, следует использовать наиболее общий способ. Если все способы решения являются общими, следует выбрать самый быстрый. Если заранее не известно, какой способ является самым быстрым, можно выполнить некоторые тесты, чтобы это узнать.

В этом разделе будут представлены два способа копирования файла на Python 3. Первый способ реализован в файле **copy1.py**:

```
#!/usr/bin/env python3
from shutil import copyfile
import sys
if len(sys.argv) >= 3:
    source = str(sys.argv[1])
    destination = str(sys.argv[2])
else:
    print('Не хватает аргументов!')
    sys.exit(0)
try:
    copyfile(source, destination)
except IOError as e:
    errno, strerror = e.args
    print("I/O error({0}): {1}".format(errno, strerror))
```

Функция `copyfile()` модуля `shutil` делает за нас всю работу. Поскольку в этом способе используется единственный метод копирования, мы никак не можем контролировать его. Иногда это хорошо, но иногда плохо. Второй способ реализован в файле **copy2.py**:

```
#!/usr/bin/env python3
import sys
if len(sys.argv) >= 3:
```

**Скорая помощь** 

Исходный код для этого урока можно найти на **LXF DVD** или в разделе Archives для **LXF227** на сайте [linuxformat.com](http://linuxformat.com) — он доступен всем.

## Утилита командной строки `strace`

Утилита `strace` позволяет узнать, что происходит внутри системы при выполнении программы или скрипта. Строго говоря, `strace` отслеживает системные вызовы и сигналы выполняемой программы и выводит их на экран. При запуске `strace` для файла **руср.py** мы получим такой результат:

```
$ strace ./руср.py руср.py anotherFile
...
stat("/usr/lib/python3.4/os.py", {st_mode=S_IFREG|0644, st_size=33763, ...}) = 0
...
open("./руср.py", O_RDONLY) = 3
...
```

```
open("anotherFile", O_WRONLY|O_CREAT|O_TRUNC|O_CLOEXEC, 0666) = 4
...
write(4, "#!/usr/bin/env python3\n\nimport o...", 2140) = 2140
close(4) = 0
close(3) = 0
...
```

Этот результат показывает, что скрипт **руср.py** выполняется в виде функций и системных вызовов C, поскольку это единственный способ связи с ядром Linux. Тем не менее, для написания той же программы на C потребовалось бы гораздо больше строк кода и часов отладки.

» Подпишитесь на печатную или электронную версии на [www.linuxformat.ru/subscribe!](http://www.linuxformat.ru/subscribe!)



```

source = str(sys.argv[1])
destination = str(sys.argv[2])
else:
    print('Не хватает аргументов!')
    sys.exit(0)
size = 16384
try:
    with open(source,'rb') as f1:
        with open(destination,'wb') as f2:
            while True:
                buf = f1.read(size)
                if buf:
                    n = f2.write(buf)
                else:
                    break
except IOError as e:
    errno, strerror = e.args
    print("I/O error(0): {1}".format(errno,strerror))

```

Этот вариант использует метод `write()` для копирования данных в новый файл с использованием буферной переменной `buf`. Использование очень маленького буфера, например, емкостью в 8 байт, поставит под угрозу производительность скрипта. Решение заключается в использовании относительно больших буферов в зависимости от объема ОЗУ вашего компьютера.

Заметьте, что в файлах `copy1.py` и `copy2.py` для перехвата и обработки потенциальных ошибок во время операции копирования используется блок `try`. Хотя оба метода хороши, второй дает больший контроль над процессом копирования и может быть эффективнее; поэтому в нашей реализации `cp` на Python 3 мы будем использовать именно его.

## Варианты обработки

Самое важное, что следует учитывать при реализации программы `myscp.py` — это ваши действия, если целевой файл уже существует. Проще говоря, собираетесь ли вы удалить целевой файл или остановить процесс копирования?

По умолчанию утилита `cp` удаляет целевой файл, если не указан ключ `-n`. Кроме того, если указан ключ `-i`, `cp` спросит, хотите ли вы удалить целевой файл. Но если используются оба ключа `-n` и `-i`, то `-n` будет иметь приоритет, а `-i` будет проигнорирован. Вторая задача — определить, является ли целевой файл каталогом, поскольку тогда надо будет поместить новый файл в этот каталог.

Здесь мы обрабатываем аргументы командной строки и параметры, связанные с копированием файлов, с помощью кода Python 3 из файла `processOptions.py`, например:

```

#!/usr/bin/env python3
import os
import sys
iSwitch = 0;
nSwitch = 0;
if len(sys.argv) == 3:
    source = str(sys.argv[1])
    destination = str(sys.argv[2])
elif len(sys.argv) == 4:
    source = str(sys.argv[1])
    destination = str(sys.argv[2])
    option1 = str(sys.argv[3])
    if option1 == "-i":
        iSwitch = 1;
    if option1 == "-n":
        nSwitch = 1;
elif len(sys.argv) >= 5:

```

```

source = str(sys.argv[1])
destination = str(sys.argv[2])
option1 = str(sys.argv[3])
option2 = str(sys.argv[4])
if (option1 == "-i" or option2 == "-i"):
    iSwitch = 1;
if (option1 == "-n" or option2 == "-n"):
    nSwitch = 1;
else:
    print('Синтаксис: ', sys.argv[0], ' источник цель [-in]')
    sys.exit(0)
# Если источник не файл
if os.path.isfile(source) == False:
    print(source, '- не файл. Выходим...')
    sys.exit(0)
# Если цель каталог
if os.path.isdir(destination):
    destination = destination + '/' + source
    print('Копирование в ', destination)
# Если цель уже существует
if os.path.isfile(destination):
    print(destination, ' уже существует!')
# Если ключ -n равен OFF, проверить ключ -i
if nSwitch == 0:
    if iSwitch == 1:
        answer = input('[Yn]: ')
        if answer == 'y' or answer == 'Y':
            print('Копирование ', source, ' в ', destination)
        else:
            print('Нельзя перезаписывать ', destination)
            sys.exit(0)
# Если ключ -n равен ON, остановить программу
elif nSwitch == 1:
    sys.exit(0)
else:
    print('Копирование', source, ' в ', destination)

```

Для простоты кода исходный файл всегда должен быть первым, а целевой файл или каталог должны быть вторыми — об этом говорится в сообщении о синтаксисе, которое выводит скрипт.

Затем можно указать любые необходимые параметры. При запуске файла `processOptions.py` мы получим следующий результат:

```

$ ./processOptions.py
Синтаксис: ./processOptions.py источник цель [-in]
$ ./processOptions.py afile
Синтаксис: ./processOptions.py источник цель [-in]
$ ./processOptions.py afile anotherFile
afile 0 это не файл. Выходим...
$ ./processOptions.py processOptions.py newFile
Копирование processOptions.py в newFile
$ mkdir test
$ ./processOptions.py processOptions.py test
Копирование в test/processOptions.py
Копирование processOptions.py в test/processOptions.py
$ touch test/processOptions.py
$ ./processOptions.py processOptions.py test
Копирование в test/processOptions.py
test/processOptions.py уже существует!
$ ./processOptions.py processOptions.py test -n
Копирование в test/processOptions.py
test/processOptions.py уже существует!
$ ./processOptions.py processOptions.py test -i
Копирование в test/processOptions.py

```

### Скорая помощь



Системный вызов — это просто функция C. Отличие системного вызова от обычной функции состоит в том, что в системных вызовах реализованы функции низкого уровня, и в том, что системные вызовы связывают ядро UNIX с остальным миром. В конечном итоге команды Python 3, связанные с файловыми операциями, преобразуются в системные вызовы C.

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

```
test/processOptions.py уже существует!
[yln]: y
Копирование processOptions.py в test/processOptions.py
$ ./processOptions.py processOptions.py test -i
Копирование в test/processOptions.py
test/processOptions.py уже существует!
[yln]: n
Нельзя перезаписывать test/processOptions.py
```

Хотя **processOptions.py** не запускает операцию копирования файлов, этот файл близок к тому, что хотим сделать мы, поэтому примем код файла **processOptions.py** за основу для **руср.py**.

## Окончательный вариант

В этой версии объединяются все предыдущие фрагменты кода и аспекты для получения конечного результата. Окончательная версия реализации утилиты *ср* на Python сохранена в файле **руср.py** и приведена ниже:

```
#!/usr/bin/env python3
import os
import sys
def fileCopy(source, destination):
    size = 16384
    try:
        with open(source, 'rb') as f1:
            with open(destination, 'wb') as f2:
                while True:
                    buf = f1.read(size)
                    if buf:
                        n = f2.write(buf)
                    else:
                        break
    except IOError as e:
        errno, strerror = e.args
        print("I/O error({0}): {1}".format(errno, strerror))
def main():
    iSwitch = 0;
    nSwitch = 0;
    if len(sys.argv) == 3:
        source = str(sys.argv[1])
        destination = str(sys.argv[2])
    elif len(sys.argv) == 4:
        source = str(sys.argv[1])
        destination = str(sys.argv[2])
        option1 = str(sys.argv[3])
        if option1 == "-i":
            iSwitch = 1;
        if option1 == "-n":
            nSwitch = 1;
    elif len(sys.argv) >= 5:
        source = str(sys.argv[1])
```

```
destination = str(sys.argv[2])
option1 = str(sys.argv[3])
option2 = str(sys.argv[4])
if (option1 == "-i" or option2 == "-i"):
    iSwitch = 1;
if (option1 == "-n" or option2 == "-n"):
    nSwitch = 1;
else:
    print('Синтаксис: ', sys.argv[0], ' источник цель [-in]')
    sys.exit(0)
# Если источник - не файл
if os.path.isfile(source) == False:
    print(source, '- не файл. Выходим...')
    sys.exit(0)
# Если цель - каталог
if os.path.isdir(destination):
    destination = destination + '/' + source
# Если цель уже существует
if os.path.isfile(destination):
    # Если ключ -n равен OFF, проверить ключ -i
    if nSwitch == 0:
        if iSwitch == 1:
            answer = input('[yln]: ')
            if answer == 'y' or answer == 'Y':
                fileCopy(source, destination)
        else:
            sys.exit(0)
    # Если ключ -n равен ON, остановить программу
    elif nSwitch == 1:
        sys.exit(0)
    else:
        fileCopy(source, destination)
if __name__ == '__main__':
    main()
else:
    print("Это отдельная программа, а не модуль!")
    Взглянув на файлы руср.py и processOptions.py, вы поймете их сходство, но также и их основные различия — в файле руср.py используется две функции и переменная __name__.
    При разработке системных инструментов тестирование имеет не менее важное значение, чем сама разработка, так как в результате тестирования мы можем определить, что программа работает правильно. Необходимо выполнить следующие проверки:
    $ ./руср.py руср.py test/
    $ ./руср.py руср.py /tmp/anotherFile
    $ ./руср.py руср.py /tmp/anotherFile -i
    ...
    $ ./руср.py
    Синтаксис: ./руср.py источник цель [-in]
```

Отлично! Теперь вы — программист для Linux! 

## Разрешения файлов в UNIX

У каждого файла в Linux есть разрешения. Взгляните на следующий вывод:

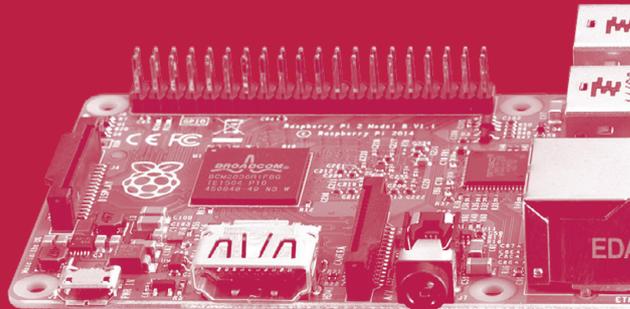
```
$ ls -l myCP.pl data.txt
-rw-r--r-- 1 mtsouk mtsouk 158 Oct 18 10:35 data.txt
-rwxr-xr-x 1 mtsouk mtsouk 158 Apr 23 2015 myCP.pl
```

В первом столбце предыдущего вывода команды `ls` показаны разрешения для каждого файла. Как видите, разрешения занимают 10 разрядов, которые также можно рассматривать как биты. Если отбросить первый бит, который определяется Linux

и задает тип записи, оставшиеся девять бит можно объединить в три группы по три бита в каждой. Первая группа — это разрешения пользователя, вторая — разрешения основной группы Linux, к которой принадлежит пользователь, и третья — разрешения остальных пользователей. Если вы знакомы с двоичной арифметикой, то легко поймете, что каждая группа может принимать значения от 000 до 111, которые в десятичной арифметической системе имеют значения от 0 до 7. Таким образом,

`rw-r--r--` также можно записать как 644, поскольку `rw` будет равно 110 (или 6), а `r--` равно 4. Аналогично, `rwxr-xr-x` можно представить как 755.

Существуют различные типы разрешений файлов UNIX. К ним, например, относятся файлы, которые не могут быть изменены или доступны текущему пользователю, поскольку они принадлежат другому пользователю. Если у вас есть права доступа к файлу, вы можете изменить текущие права доступа к этому файлу командой `chmod`.



Ваша порция смачных новостей, обзоров и учебников от Raspberry Pi

**РЕЙЧЕЛ ВОНГ**  
— аспирантка,  
желающая соз-  
давать готовые  
органы.



Привет...

Все дни я провожу в лаборатории, выращивая клетки глаза с целью изучения детской слепоты. Но я люблю мечтать и думать о будущих открытиях, что изменят мир к лучшему. Я заметила, что сейчас жутко не хватает междисциплинарных исследований, и подумала: «А если попробовать самой преодолеть разрыв между гуманитаристикой, наукой и технологией?» Так я решила заняться кодированием и робототехникой. Моя история с Raspberry Pi началась с выхода Pi Zero W. На той же неделе я успешно перепутала в Твиттере Pi Hut с модтури и Pimoroni и оказалась на пятом дне рождения Raspberry Pi в Кембридже! С этой ошибки и началась моя дружба с сообществом Raspberry Pi. Училась у энтузиастов Pi я буквально на лету, посещая разные Raspberry Jam'ы по всей стране. Когда я нуждалась в помощи, стоило написать в Твиттер, и тут же какая-нибудь добрая душа решала все мои проблемы. Моим первым проектом и подлинным самовыражением был робот Tinu 4WD, украшенный My Little Ponies. Положительные отзывы вдохновили меня на создание технических безделушек, носимых на себе. Употребив свои умения в изготовлении бижутерии, я сделала корону из светодиодов с несколькими режимами и светящуюся открытку Unicorn rHat. Потом сделала еще серьги, которые тоже по-разному светятся. Прямо сейчас я экспериментирую, внедряя в такие аксессуары Chibitronics, музыку и BBC micro:bit. Планирую создать свой клуб Code & Craft в Лондоне, буду учить кодировать на изготовлении бижутерии или открыток.

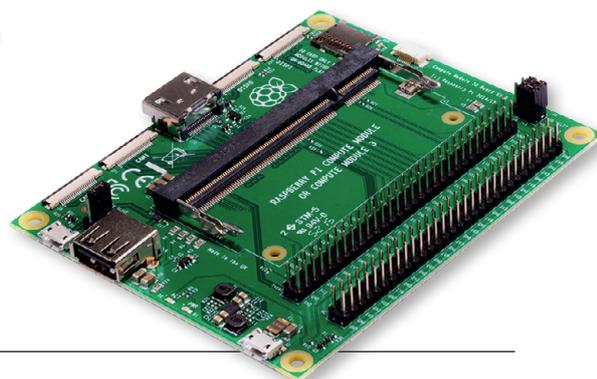
## Ubuntu Core на Pi3 Compute Module

Canonical берет Pi в оборот — в хорошем смысле.

Raspberry Pi Compute Module 3 (CM3), близнец Pi 3 по аппаратной части, теперь может использоваться с Ubuntu Core от Canonical, облегченным собратом Ubuntu. Серия CM Pi относится к недорогим чипам DDR-2 и была разработана с расчетом как на любителей, так и для индустрии. CM3 добавляет вычислительной мощности и в сочетании с легко устанавливаемыми «снимками [snap]» Ubuntu Core представляет собой идеальный недорогой компьютер общего назначения.

Canonical поддерживает инициативу Software Defined Everything (SDx), предполагающей, что устройства IoT, как и все прочие, должны работать от полноценных операционных систем, а не от специализированных коммерческих (и, скорее всего, небезопасных), и у такого союза найдется немало поклонников и применений. Поскольку формат

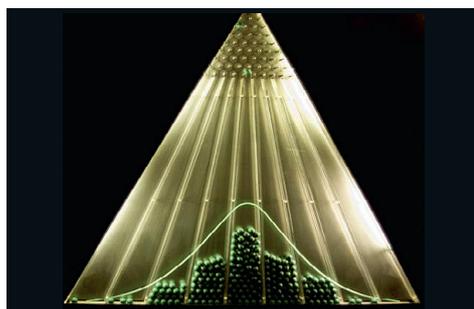
универсальных снимков наращивает популярность, Pi 3 становится еще привлекательнее для разработчиков IoT, ищущих безопасные и экономичные решения. Подробности здесь: <https://insights.ubuntu.com/2017/06/27/canonical-supports-ubuntu-core-on-the-raspberry-pi-compute-module>



## Игра в шарики

Наука, статистика, сенсоры.

Специалисты по субатомарной физике в университете Дарема [Durham] создали свой вариант доски Гальтона на базе Pi — научную демонстрацию, где шарики падают и, проходя между штырьками, распределяются по Гауссу вокруг центра. Она представлена на Летней научной выставке Королевского общества. [www.raspberrypi.org/blog/royal-society-galton-board](http://www.raspberrypi.org/blog/royal-society-galton-board)

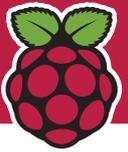


## Picorder

Видеозапись на ладони.

Уэйн Кинан [Wayne Keenan] создал миниатюрный видеорегистратор, используя Pi Zero W, его модуль камеры, экран Pimoroni HyperPixel и аккумулятор LiPo. Управляется видеорегистратор пока через *raspivid* по SSH, но поскольку у экрана имеются сенсорные функции, для него можно создать приличный графический интерфейс. <http://bit.ly/2u0b33o>





# Thonny

Лес Паундер добрый месяц тестировал альтернативную среду разработки Python, так что не *IDLE*’ом единым.

## Вкратце

» *Thonny* — альтернативный редактор Python для всех дистрибутивов Linux. Имеет функциональный, но простой интерфейс, содержащий всё, чтобы помочь новичкам разбираться в коде, который они пишут. А объяснение ошибок и принципов работы кода делает его больше чем просто редактором Python.

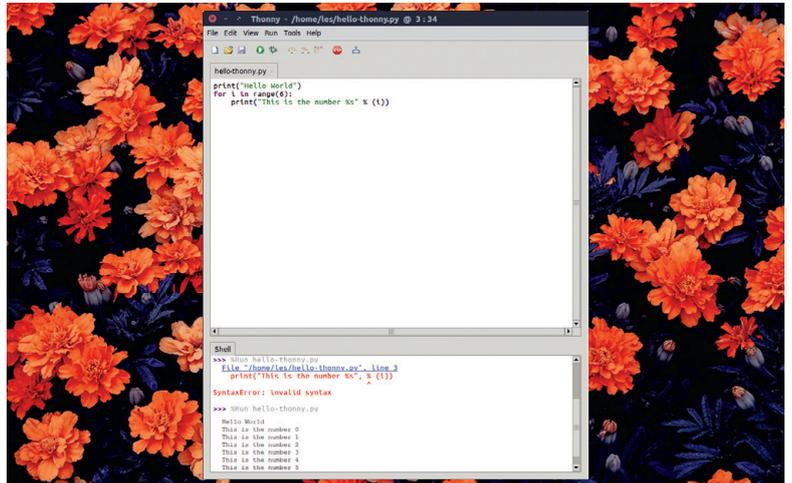
Появление Raspberry Pi добавило Python популярности. Всё больше людей сегодня начинают осваивать Python, но редактор по умолчанию *IDLE*, при всей своей функциональности, кажется новичкам громоздким и неповоротливым. Так что, узнав о выходе *Thonny*, IDE Python для начинающих, мы просто не могли этого пропустить...

*Thonny* — это несложная среда разработки для Python 3.6. Здесь предлагается единое окно с панелью редактирования поверх оболочки Python (REPL). Для работы с несколькими файлами используются вкладки — это намного удобнее, чем несколько окон. Код, написанный в редакторе, можно запустить и увидеть результат в оболочке; при возникновении проблем оболочка точно определит их источник.

Мы протестировали это на ошибке форматирования строк, поставив апостроф там, где он не требовался. *Thonny* выявил ошибку синтаксиса и указал на проблему. И это приводит нас к еще одной отличной функции: вызвав простой отладчик (Ctrl+F5), можно пройти по коду небольшими шагами (F7), наблюдая, как Python с ним работает и как оценивает наши выражения. Можно также пройти по строкам, нажав F6.

Как и в других IDE, в *Thonny* действует автодополнение кода, чтобы пользователь мог изучить полный список классов/функций, принадлежащих библиотеке, и быстро написать код Python. Но, уже не как у других, здесь всё происходит только в одном окне, а не в их множестве.

*Thonny* создает свою собственную среду Python, отдельную от той, что установлена



» *Thonny* привлекает простым и понятным интерфейсом, и под ним скрывается мощный редактор Python, готовый принять пользователя.

в системе, и пользователь может устанавливать любые библиотеки, не опасаясь конфликтов. Очень полезная функция для любителей поковыряться во внутренностях. И к слову об установке ПО: в *Thonny* есть собственный графический интерфейс к менеджеру пакетов Python (*pip*), для установки библиотек в среде Python от *Thonny*.

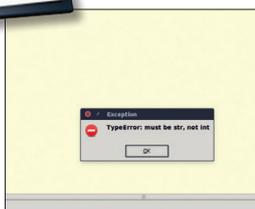
*Thonny* также работает с *micro:bit* — почти по тому же принципу действия, что и *mu* от Python Software Foundation. Чтобы использовать *Thonny* с *micro:bit*, надо установить плагин, а затем сменить интерпретатор из Python 3.6 *micro:bit* через меню Tools > Options [Инструменты > Параметры]. Тогда вы сможете подключать *micro:bit* перед тем как запустить код или взаимодействие с ним через оболочку Python (REPL).

Чтобы установить *Thonny* на вашу систему, надо загрузить скрипт оболочки, сделать его исполняемым и запустить. И это не лучший способ для упаковки приложения, поскольку он полагается на доверие к источнику. Далее вы можете выполнить чтение скрипта установки, который автоматически загрузит пакет для вашей архитектуры процессора, а затем извлечет файл. А еще можно установить *Thonny* через *pip*, это более безопасная альтернатива скрипту оболочки.

Невзирая на эти проблемы, *Thonny* — похвальная альтернатива *IDLE*. У него простой дружелюбный интерфейс, а внутри готов мощный арсенал плагинов. Будем надеяться, имеющиеся шероховатости будут исправлены в следующем релизе. LXF

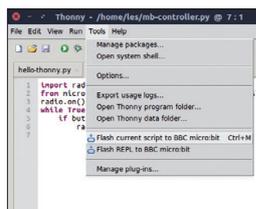


## Свойства навскидку



### Простой отладчик

Пройдитесь по своему коду, найдите источники всех проблем и не оставьте шансов никаким ошибкам синтаксиса!



### Гармония *micro:bit*

Редактирование *MicroPython* в том же редакторе, что и обычный Python — идеально для любителей *Raspberry Pi*.

## Не без огрехов

*Thonny* не идеален. По факту — именно потому, что среда Python здесь существует отдельно от основной системы, вам надо либо установить обычные пакеты *Raspberry Pi* (RPi.GPIO, GPIO Zero, Sense HAT), либо переключить интерпретатор на Python, установленный в системе. Но тогда нельзя устанавливать программы через встроенный графический интерфейс. Если в следующих релизах это будет исправлено, мы полагаем, *Thonny* имеет хорошие шансы стать редактором Python по умолчанию на *Raspberry Pi*.

Еще одна проблема в том, что *Thonny* доступен для всех дистрибутивов Linux только через его собственный установщик.

## LINUX FORMAT Вердикт

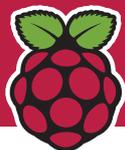
### Ubuntu MATE 16.04.2 LTS

Разработчик: Айвар Аннамас [Aivar Annamaa]  
Сайт: <http://thonny.org>  
Лицензия: MIT

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

» *Thonny* — простая, но мощная альтернатива *IDLE*. Идеален для решения общих задач в Python.

## Рейтинг 7/10



# Node-RED: Система оповещения

Лес Паундер поможет нам освоиться с Node-RED, создав в Twitter систему оповещения по хэштегу, использующую визуальные и звуковые сигналы.



**Наш эксперт**

**Лес Паундер** — свободный хакер и умелец, который любит возиться с техникой. Он поддерживает проект PISCADemy от Raspberry Pi в качестве тренера и ведет блог на <http://www.bigl.es>.

## Вам нужны

- » 40-контактный Raspberry Pi любой модели.
- » Raspbian Pixel
- » Подключение Pi к Интернету
- » Макетная плата
- » Светодиод
- » 1 переходник мама-папа
- » 1 резистор 330 Ом (оранжевый, оранжевый, коричневый и золотой).
- » Ноутбук или настольный ПК
- » Получить код: <https://github.com/lesp/LXF227-Node-RED/archive/master.zip>

**Н**а этом уроке мы познакомимся с Node-RED, мощным визуальным языком, в котором код состоит из соединенных между собой «узлов». Мы создадим систему оповещения через Twitter, которая будет сообщать нам о появлении твитов с определенным хэштегом. При получении твита будет загораться лампочка, а затем сам твит сохранится в файл. Далее он будет прочитан пользователю с помощью речевого преобразователя [text-to-speech]. В случае нашего отсутствия, оповещение будет отправлено по электронной почте.

Из аппаратных средств для этого проекта нам нужен одиночный светодиод, подключенный к GPIO21 и к земле [GND] через резистор 330 Ом. Подробнее см. на схеме в материалах для скачивания к этому проекту.

Загрузив Raspberry Pi и рабочий стол Raspbian, щелкните на главном меню. Теперь перейдите в меню Programming [Программирование] и нажмите Node-RED, чтобы запустить сервер Node-RED. Откроется терминал с прокруткой информации. В терминале будет выведен IP-адрес, как правило, 192.168.X.X:1880. Это адрес нашего Raspberry Pi в сети; за ним идет номер порта, открываемого Node-RED. Теперь откройте на своем ноутбуке или настольном компьютере новое окно браузера и введите этот IP-адрес в адресную строку.

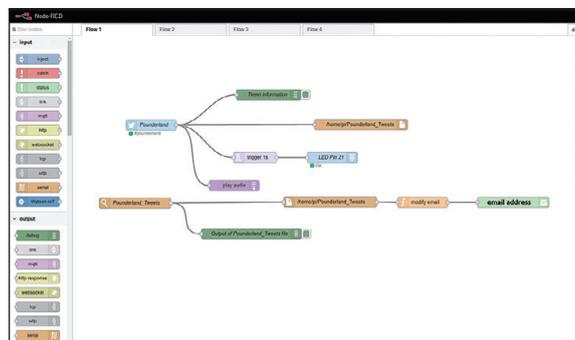
Интерфейс Node-RED разделен на три столбца. Узлы [nodes] — это блоки, которые мы используем для создания кода; блоки перетаскиваются в центральный столбец, где находятся Потoki [flows], которые составляют код проекта. В правом столбце содержится информация об отдельных узлах, или можно просматривать там код отладки.

## Создание кода

Давайте построим первую часть нашего кода. Это будет узел Twitter, который свяжет наш код с Twitter. Прокрутите столбец узлов до Social [Общение], а затем перетащите узел Twitter в столбец Поток. Наш узел Twitter содержит два символа. Синий круг сообщает, что узел не был развернут, а красный треугольник указывает, что узел надо настроить. Щелкните дважды по самому узлу, а затем по значку карандаша, чтобы войти в свою учетную запись Twitter и предоставить Node-RED право ее использовать.

Затем вернитесь к редактированию узла Twitter. Ваш идентификатор Twitter должен быть заполнен, поэтому убедитесь, что в поле Search [Поиск] выбрано All Public Tweets [Все общедоступные твиты]. В поле For [По] введите хэштег для поиска. Мы использовали #rounderland. В поле Name [Имя] дайте узлу заголовок, например имя хэштега, который вы ищете. Если вы нажмете Deploy [Развернуть], синий кружок исчезнет, но может появиться надпись Rate Limit Reached [Достигнут лимит скорости] — это означает, что нам нужно подождать, прежде чем мы сможем использовать Twitter.

Теперь посмотрим вывод из Twitter. В списке узлов вывода найдите Debug [Отладка]. Перетащите этот узел в Поток, а затем



» Интерфейс у Node-RED на редкость аккуратный, порой кажется сложноватым, но можно отфильтровать узлы с помощью поиска, левый верхний угол.

щелкните и удерживайте серый круг узла отладки. Подключите его к узлу Twitter. Дважды щелкните узел Debug / msg.payload, чтобы отредактировать его свойства. Вывод / "payload" — содержимое твита; оно отправляется на вкладку отладки, это крайний правый столбец на экране.

Далее следует узел Tweet Information. Он будет выводить твиты, соответствующие нашему хэштегу, на вкладку отладки. Снова нажмите кнопку Deploy, и спустя несколько минут отправьте твит с заданным хэштегом. Теперь вернитесь к Node-RED и взгляните во вкладку отладки. Вы увидите, что там появился твит, а значит, теперь мы можем получать данные для нашего проекта.

Далее сохраните твит во внешнем файле. Прокрутите вниз до раздела Storage [Сохранить в...] и выберите узел File [Файл] с иконкой файла справа от узла. Перетащите его в Поток и подключите к узлу Twitter. Теперь дважды щелкните узел файла и создайте новое имя файла, включая полный путь! В нашем случае это /home/pi/pounderland\_tweets, и если файл еще не существует, он создастся. В поле Action [Действие] должно быть выбрано Append to File [Добавить в файл]. Теперь назовите узел, указывающий путь к файлу. Опубликованные твиты будут добавляться в файл.

## Зажгите лампочку

Теперь надо добавить светодиод, который будет уведомлять нас о получении твита. Из узлов Function [Функции] перетащите узел Trigger [Триггер] и подключите его к узлу Twitter. Дважды щелкните по Trigger и измените время ожидания до одной секунды. Тогда в течение 1 секунды функция будет иметь значение 1, а затем отправит значение 0. Теперь измените имя узла на Trigger 1s.

Затем подключите светодиод к триггеру. Из узлов Raspberry\_Pi выберите узел gpio gpio, с логотипом Pi справа. Перетащите его в Поток и подключите к Trigger 1s. Теперь дважды щелкните по узлу gpio и измените выход GPIO на Pin 40 — GPIO 21. Обеспечьте, чтобы в поле Type [Тип] было указано Digital Output [Цифровой выход].



## Азы Node-RED

Node-RED — это браузерный редактор, который использует узлы, подключенные к потоку, для связи онлайн-мира с физическим, обычно именуемым Интернетом Вещей — IoT (Internet of Things). Редактирование потока осуществляется за счет подключения узлов.

Каждый узел помечен серыми точками. У одних они расположены слева — это узлы вывода, например, для отображения данных в консоли отладки. У других — справа, это узлы ввода, такие

как подписка на данные через mqtt. Некоторые узлы имеют серые точки с обеих сторон, например, Delays [Задержки], которые могут применяться для замедления потока.

Другие узлы можно редактировать для выполнения различных задач, например, узел Function, который используется для написания пользовательских функций в JavaScript. Этот узел также может иметь несколько выходов, каждый из которых несет свою информацию.

Но насколько мощным является Node-RED?

Д-р Люси Роджерс [Lucy Rogers], судья *Robot Wars* [Войны роботов] на ТВ BBC и невероятный хакер и умелец, использует Node-RED, чтобы воскрешать динозавров на тайном острове... извините, мы хотели сказать, она помогает парку развлечений вернуть к жизни свои старые экспонаты динозавров, создавая с помощью Node-RED интерактивную сеть динозавров, которые реагируют на разные сигналы от посетителей и других экспонатов.

Затем дайте узлу имя LED Pin 21. Нажмите Done [Готово], чтобы сохранить. Когда значок предупреждения ограничения скорости исчезнет, нажмите Deploy. Отправьте твит с нужным хэштегом, и увидите, что лампочка загорелась.

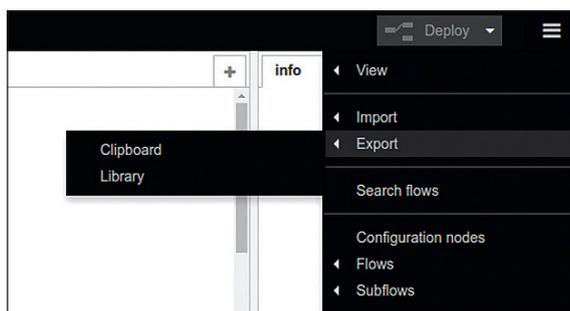
## Говорящие твиты

Чтобы добавить последний в этом потоке узел, зайдите в категорию Output [Выход] и перетащите узел Play Audio [Воспроизвести аудио]. Подключите его к узлу Twitter. Дважды щелкните по узлу Play Audio и измените голос TTS (Text To Speech) на Google UK English Female [Британский английский, женский], затем нажмите Done. Затем щелкните по Deploy, дождитесь окончания ограничения скорости и отправьте твит. Теперь вы должны услышать свой озвученный твит!

Давайте создадим второй поток. На сей раз нам нужен узел из категории Advanced [Дополнительные], и называется он Watch [Наблюдение]. Этот узел наблюдает за статусом файла или папки и запускается, когда видит изменения. Перетащите узел Watch в Поток и измените поле File на путь к файлу, куда добавляются твиты. Дайте узлу имя. Затем перетащите узел Debug из Output и подключите его к узлу Watch. Дайте узлу отладки имя, указывающее, что он будет отображать вывод файла в консоли отладки.

Теперь подготовим файл для отправки по e-mail. В категории узлов Storage [Расположение] найдите файл с серыми точками справа и слева. Перетащите его и подключите к узлу Watch. Имя файла измените, чтобы он соответствовал пути и имени вашего файла.

Следующий блок задает строку темы нашего письма, с помощью узла Topic of the message [Тема сообщения]; в данном случае это имя файла. Из узлов Function [Функция] перетащите Function в Поток и подключите ее к предыдущему узлу файла. Затем



➤ Справа сверху расположено меню Node-RED, предназначенное для развертывания потоков и экспорта в JavaScript.

дважды щелкните по узлу Function, назовите его Modify email [Изменить сообщение], а затем введите следующий код, чтобы изменить название файла в имени по своему усмотрению. По завершении нажмите Done.

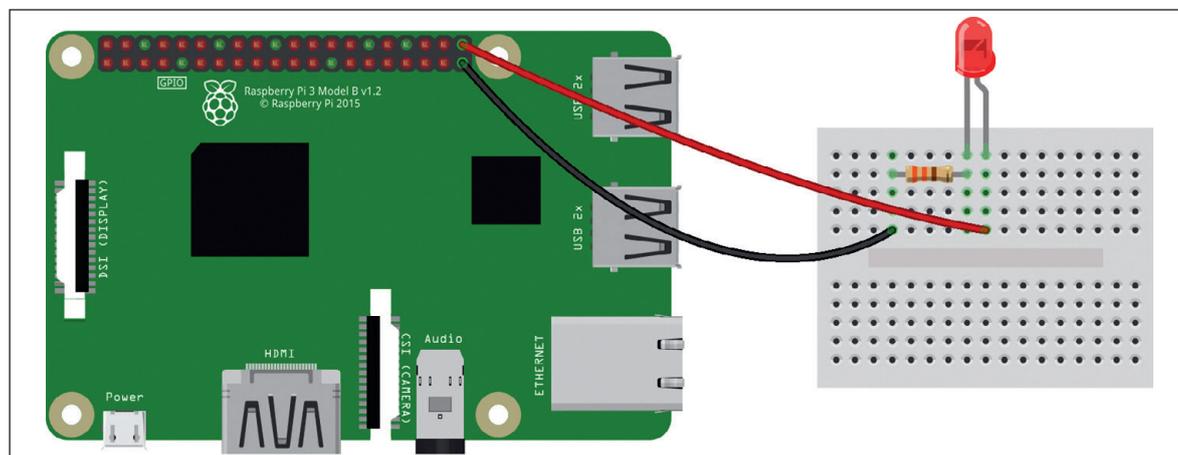
```
msg.topic = msg.topic.replace("/home/pi/pounderland_tweets",
"Ух ты, вот это новость!!!");
return msg;
```

Наш последний узел — это Email из категории Social. Перетащите его и подключите к узлу Modify Email. Введите данные своего электронного адреса от интернет-провайдера или электронной почты. Теперь нажмите Done, дождитесь отмены лимита скорости, а затем проверьте свой новый проект.

Вы сделали свои первые шаги с Node-RED и создали систему оповещения о твитах с обратной связью в виде светового сигнала, преобразования текста в речь и сохранения в файл, и всё это с минимальным объемом кода. LXF

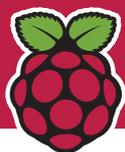
## Скорая помощь

Вместо того, чтобы подключать Pi к экрану, клавиатуре или мыши, можно запускать Node-RED через SSH. Для этого надо только войти в Pi через SSH (на этот счет есть куча руководств). А затем в терминале ввести `$node-red-start`, чтобы запустить сервер Node-RED.



➤ Схема этого проекта проста: нужны всего два элемента Raspberry Pi и два провода.

» Подпишитесь на печатную или электронную версии на [www.linuxformat.ru/subscribe!](http://www.linuxformat.ru/subscribe!)



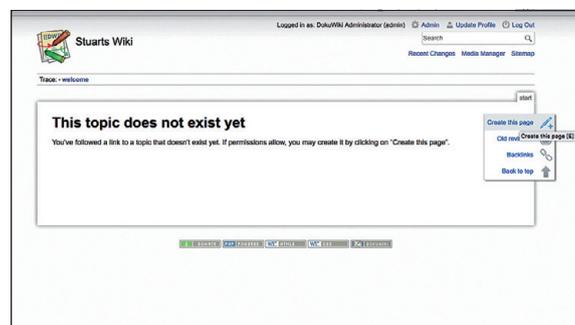
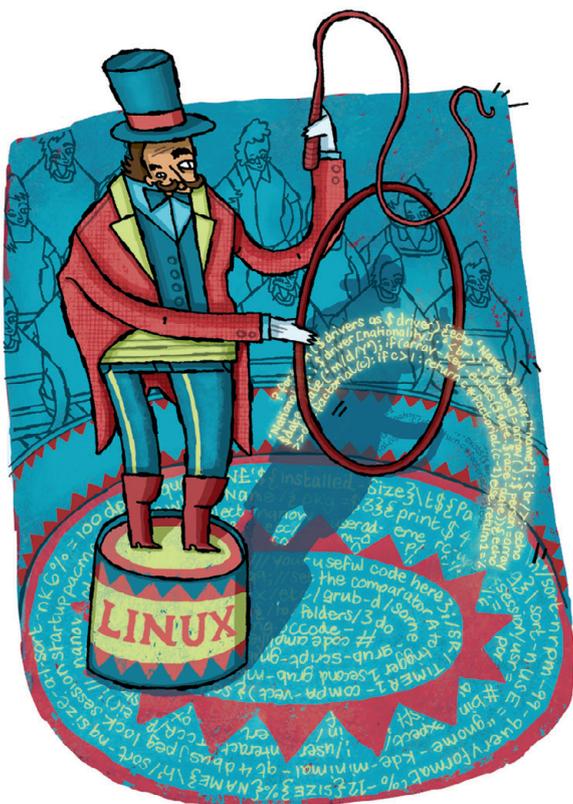
# Wiki: Знания без проводов

Стюарт Бернс показывает, как создать беспроводной сервер wiki на крошечном компьютере Raspberry Pi W и спрятать его в дальнем углу.



Наш эксперт

Стюарт Бернс работает системным администратором в компании Fortipe 500. Он специализируется на не-Wintel платформах, а в свободное время часто возится с чем попало, относящимся к технологиям.



➤ Создавать новые страницы в wiki легко.

Первое, что вам надо сделать — создать загрузаемую карту MicroSD с образом Raspbian OS. Из-за ограниченной природы нашей платформы мы воспользуемся изданием Raspbian Lite, а не полной нафаршированной настольной версией. Образ Raspbian Lite можно найти на [https://downloads.raspberrypi.org/raspbian\\_lite\\_latest](https://downloads.raspberrypi.org/raspbian_lite_latest). Мы будем использовать прямое скачивание, а не торрент-файл. Скачайте ZIP-файл на свой ПК.

Когда вы откроете ZIP-файл, надо будет извлечь только файл образа (с окончанием на .img). Некоторые инструменты для извлечения архива еще и открывают файл образа. Убедитесь, что по окончании процесса у вас только один файл образа.

Хотя можно записать образ на карту microSD в старомодной манере — командой dd, мы рекомендуем использовать новый GUI, с единственной целью записи образов SD. Именно такой подход рекомендуют сами создатели Pi. Эта новая программа называется *Etcher*. Она свободна для скачивания и работает на всех основных платформах. *Etcher* ликвидирует сложности команды dd — это означает, что больше не будет случайной перезаписи нужной информации или возни с иными нежелательными последствиями.

## Скачиваем Etcher

Скачайте *Etcher* с <https://etcher.io> и установите версию Linux. Для ее запуска надо только распаковать приложение и дважды щелкнуть по нему — не придется тратить время на его установку через системы управления пакетами. При первом запуске приложение настроит несколько пунктов и предложит создать ярлык для дальнейшего использования. Использование *Etcher* для записи образа определенно является шагом вперед!

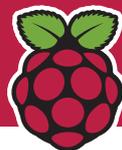
Запустив приложение *Etcher*, вы увидите только три кнопки и два пункта, которые надо настроить: исходный файл образа Raspbian и конечный (карта MicroSD). Нажмите на кнопку Source и перейдите в распакованный образ Raspbian, извлеченный ранее.

С того момента, как появился первый Raspberry Pi, было множество изменений, включая больше ядер, больше функций и больше ОЗУ. Всё это хорошо, но в новом Pi Zero W Raspberry Pi Foundation возвращается к минимализму.

Zero W полностью ориентирован на мобильные технологии в крошечной упаковке, но со встроенной беспроводной сетью (отсюда и W в названии), и всё это при стоимости около £11. И хотя Zero W, возможно, менее мощный, чем новый Pi 3, его производительность достаточно хороша для самоделкиных, т.е. он вполне соответствует изначальным этическим принципам Raspberry Pi. На этом уроке вы узнаете, как использовать Zero W для создания беспроводной wiki без клавиатуры, мыши и монитора. Это переносная wiki у вас в кармане!

Если вы планируете покупку Pi Zero W, вам потребуются дополнительные переходники для USB- и HDMI-подключений, поскольку на устройстве все порты ввода/вывода формата micro. К счастью, большинство продавцов Pi имеет запас требуемых переходников.





## Если SSH не работает как надо...

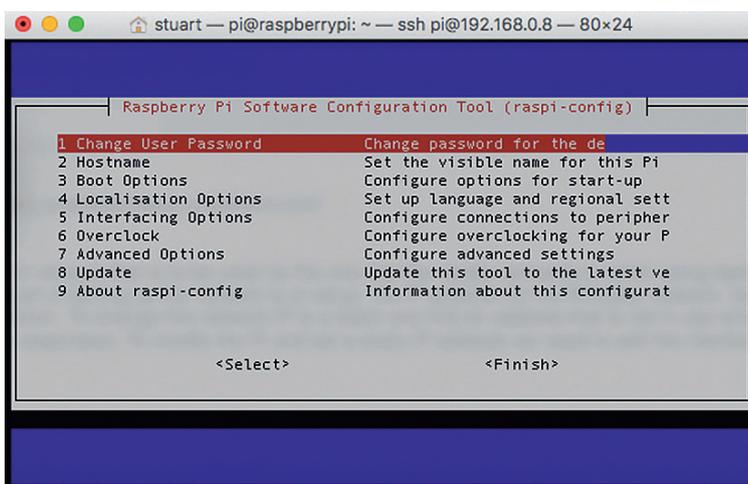
При установке Raspbian иногда возникает странная ошибка насчет отсутствия доступных алгоритмов. Проблема появится, только если пользователь попытается войти удаленно через SSH. Решить проблему входа через SSH можно, указав ключ `--vvv`. По неизвестным причинам, сервер OpenSSH при установке генерирует ключи неправильно. Решением данной проблемы будет заново создать нужные ключи вручную.

Эта ошибка появляется вместе с сообщением со строками "No supported key exchange protocols available [Недоступны протоколы поддерживаемого обмена ключей]" при отладке с `--vvv`. Проверьте отсутствие протокола, войдя в `/etc/ssh` и взглянув на закрытый и открытый ключи. Если они имеют нулевой размер, это означает, что у читателя проблема.

Необходимо пересоздать ключи для каждого протокола. В нашем случае это было исправлено такими командами:

```
ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key
ssh-keygen -t ecdsa -f /etc/ssh/ssh_host_ecdsa_key
ssh-keygen -t ed25519 -f /etc/ssh/ssh_host_ed25519_key
```

По завершении воссоздания ключей мы бы предложили также перезапустить сервер SSH и убедиться, что всё в порядке.



» Вот что надо делать, когда SSH соединение просто не работает.

Затем выберите соответствующую карту microSD в качестве целевого устройства. И, наконец, нажмите Go. Процесс записи и верификации образа займет несколько минут.

С этого момента вы уже должны быть в состоянии перенести microSD на Pi и загрузиться с нее. По умолчанию имя пользователя и пароль — "pi" и "raspbian". Изменения, вносимые на данном этапе, включают настройку беспроводного соединения, обновление ОС, настройку статической сети и установку *openSSH* для удаленного доступа.

Для отображения доступных SSID (сетей) используйте код

```
sudo iwlist wlan0 scan
```

Помимо отображения доступных SSID, вывод также доказывает, что Pi видит беспроводные сети. Первый шаг — отредактировать SSID и пароль для сети, который будет использоваться. Чтобы вручную указать пароль шифрования, отредактируйте следующий файл для автоматического включения беспроводной сети в будущем:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Вместо отображения пароля в виде простого текста можно указать беспроводную сеть и ключ шифрования и автоматически заполнить этим файл `wpa_supplicant.conf`. Создать содержимое `wpa_supplicant.conf` можно командой `wpa_supplicant`. Она эффективно создаст все данные, необходимые для установления беспроводного подключения, и передаст их на экран. Это хорошо само по себе, но полезнее будет перенаправить вывод в файл настройки. Благодаря этому можно избежать любой возможности копирования и вставки ошибок настройки:

```
sudo su -
wpa_passphrase "SSID" "Password" >> /etc/wpa_supplicant/wpa_supplicant.conf
```

Часть "SSID" означает сеть, которая должна быть использована (например, та, к которой вы подключитесь при использовании ноутбука). Паролем должен быть сетевой пароль простым текстом, использованный ранее.

## Адресок не скажете?

Настроив беспроводной аспект, мы можем переходить ко второй части настройки сети — определению IP-адреса беспроводной

сети. Наличие статического IP-адреса для серверов считается наилучшим вариантом: им намного проще управлять, и IP-адрес сохраняется при перезагрузке. Чтобы изменить сетевой IP-адрес на статический, найдите адрес, который не используется, и запишите его. Для поиска неиспользуемого IP-адреса можно скомандовать `ping` (при условии, что брандмауэры не блокируют отклики). Чтобы изменить Pi и настроить статический IP-адрес, отредактируйте файл интерфейса. Для этого обратитесь к текстовому редактору *nano*:

```
sudo nano /etc/network/interfaces.
```

Надо изменить несколько строк. Измените строку для DHCP на `static` [статический] и введите приведенные далее строки. Эти настройки должны подойти для большинства сетей, за исключением IP-адреса: вместо него введите неиспользуемый IP-адрес сети. Если соответствующая сеть другая (например, если у вас адрес `10.0.0.0/24`), то надо изменить шлюз и сетевую маску, а также IP-адрес. Как говорится, каждый прошибает стену по-своему!

```
iface wlan0 inet static
address 192.168.0.8
netmask 255.255.255.0
gateway 192.168.0.254
dns-nameservers 8.8.8.8 8.8.4.4
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```



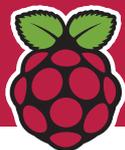
» Простой в работе инструмент *gibbons* позволит быстро редактировать страницы.

## Скорая помощь

Создание wiki может показаться чрезмерным, но это может быть полезно. Секрет в том, чтобы всё было хорошо организовано и не устарело. Если есть полезные PDF или прочие приложения, приложите их. Резервное копирование, переход и поддержка относительно просты по сравнению с бумажными заметками. Да и поиск тоже проще.

»

» Подпишитесь на печатную или электронную версии на [www.linuxformat.ru/subscribe!](http://www.linuxformat.ru/subscribe!)



## Скорая помощь



Для желающих сделать со своим Pi Zero W чуть больше есть возможность добавить плату ввода/вывода. Это позволяет взаимодействовать со внешними устройствами. Обратите внимание, что использование контактов потребует от вас умения паять.

Сохраните эти изменения с помощью Ctrl+X, и администратор предложит сохранить файл. Чтобы убедиться, что сеть чистая, на данном этапе предлагаем перезагрузиться. Если это не удастся сделать, сетевые сервисы можно перезапустить командой

```
/etc/init.d/networking restart
```

Мы также советовали бы пропинговать Pi с другого компьютера, чтобы убедиться в правильности настройки сети.

Обновите ОС командой, приведенной ниже — она заодно перезагрузит компьютер. Хотя это может выглядеть немного неуклюже, перед перезагрузкой процесс обновляет все программы до последних релизов:

```
sudo su -
apt-get update -y && apt-get upgrade -y && reboot
```

Теперь, когда базовая установка настроена и видна в сети, надо установить сервер OpenSSH для обеспечения удаленного управления Pi:

```
sudo apt-get install openssh-server -y
```

Чтобы включить интерфейс *openSSH*, читателю поможет приложение *raspi-config*. Перейдите в строку интерфейса в GUI и выберите опцию SSH. Чтобы запустить инструмент *raspi-config*, просто введите команду `sudo raspi-config` в терминале. В инструменте *raspi-config* можно изменить имя хоста с `raspberrypi` на что угодно. Затем выйдите из программы и проверьте, что доступ через SSH работает, войдя в него с другого компьютера Linux по команде

```
ssh pi@static_ip_address_just_set
```

И, наконец, займемся интерфейсом HDMI. По умолчанию система может не загружаться, если подключенный интерфейс HDMI не активирован. Так что вам придется быстро отредактировать, чтобы позволить системе загрузиться без подключения кабеля HDMI:

```
sudo su -
echo "hdmi_force_hotplug=1" >> /boot/config.txt
```

Когда это будет сделано и система загрузится без HDMI, вы сможете убрать Pi с глаз долой. Если вы столкнулись с какой-либо проблемой, присоедините кабель обратно, и всё должно загрузиться нормально. Однако вам, вероятно, придется отсоединить и заново подключить питание. Перезагрузка и удаленный вход через SSH позволит вам проверить работу системы, и вы можете настроить ее удаленно по своему усмотрению.

У OpenSSH есть проблема с правильной установкой и созданием открытого и закрытого ключей. Ее решение описывалось ранее в этой статье (см. врезку «Если SSH не работает как надо...»). Ну, а если всё работает, настало время создания wiki.



➤ Одиннадцать монеток по £1 (более или менее) предоставят вам Pi Zero W.

## Подготовка wiki

Начнем с установки *Apache* и требуемых зависимостей *Dokuwiki*, используя приведенную ниже команду, через SSH на вновь включенном удаленном сервере:

```
ssh pi@192.168.0.10
sudo apt-get install apache2 libapache2-mod-php
sudo a2enmod rewrite
service apache2 restart
```

Для проверки, что *Apache* работает, попытайтесь открыть IP-адрес сервера в браузере. Надо надеяться, что он покажет сайт *Apache* по умолчанию.

Теперь пора изменить настройку сервера *Apache* для установки wiki. Код ниже выполняет несколько простых шагов. Во-первых, он перемещается в корневую директорию сайта и скачивает последнюю версию *Dokuwiki*. Затем он распаковывает *Dokuwiki* в поддиректорию *web*-папок *Apache*, чтобы к ним можно было получить доступ внутри структуры текущей директории *http*. И меняет владельца файлов на текущего владельца (*www-data*).

```
cd /var/www
sudo wget https://download.dokuwiki.org/src/dokuwiki/dokuwiki-stable.tgz
sudo tar xvf dokuwiki-stable.tgz
sudo mv dokuwiki-*/ dokuwiki
sudo chown -R www-data:www-data /var/www/dokuwiki
```

Последний шаг — изменение корневого документа *Apache*, чтобы указать на папку *dokuwiki*. Откройте файл настройки *Apache*, следующей командой:

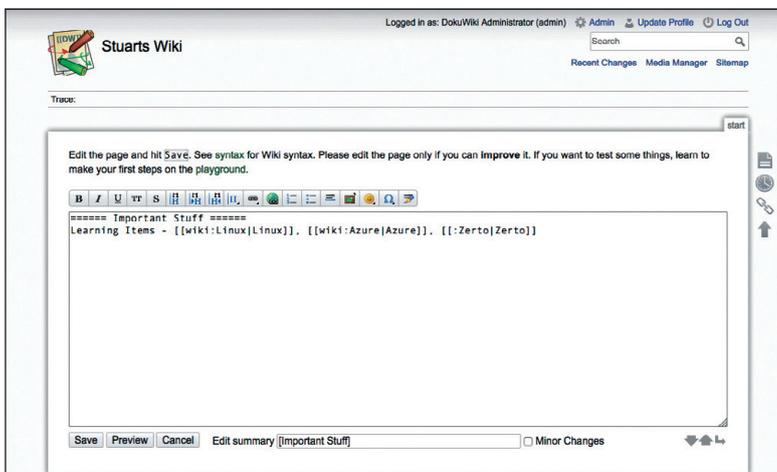
```
sudo nano /etc/apache2/sites-enabled/000*.conf
```

Определите местоположение строки *DocumentRoot* и замените ее на *DocumentRoot /var/www/dokuwiki*. Сохраните изменения и введите следующее, чтобы перезапустить сервис *Apache* для его обновления с внесенными изменениями:

```
sudo apachectl restart
```

Осталось только настроить программу wiki. Настройки *Dokuwiki* требуют всего лишь запустить основанную на *web* программу установки. Откройте браузер и перейдите на URL, показанный далее (подставив свой собственный IP-адрес). Формат должен быть таким: `http://192.168.0.10/install.php`.

Возможно, вы получите предупреждение: «Функция PHP `utf8_encode` недоступна». Решение этой проблемы сводится всего лишь к такой команде:



➤ Вот очень простая страница, ведущая на другие страницы.

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



## Pi на все случаи

Изначально был один Raspberry Pi. Теперь имеется немалое количество доступных моделей.

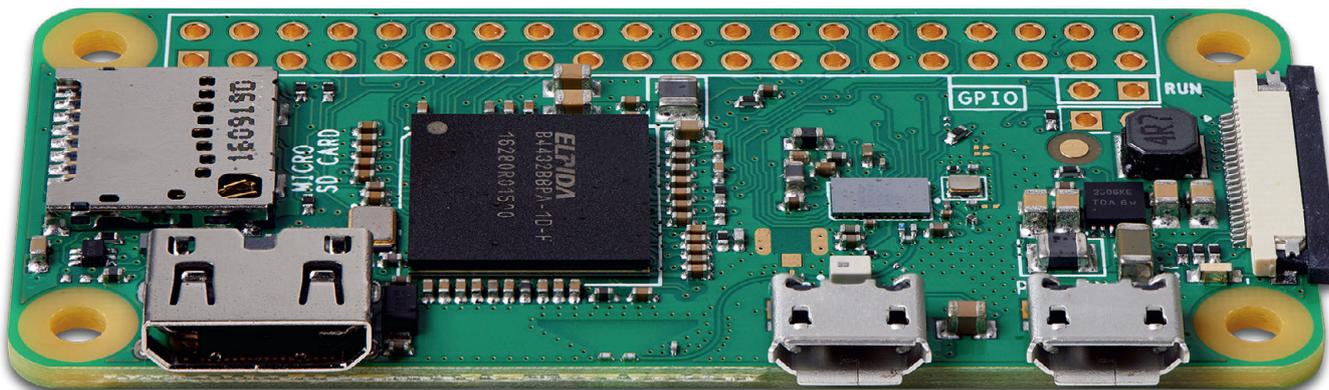
Pi Zero W, использованный на нашем уроке — это одноплатный компьютер с 1 ГГц и 512 МБ ОЗУ. Хотя такое может показаться несолидным, Zero разработан для встраиваемого устройства с очень малым энергопотреблением. Крошечный размер устройства (65×30×5мм) делает его идеальным

для проектов, требующих не огромных затрат энергии, а гибкости. Конечно, гибкость у таких крошечных маломощных плат огромная, не только в плане опций соединения, но также и по питанию устройства. Умельцы даже умудрились подключить к ним солнечные батареи и обычные батарейки.

Эта гибкость делает Pi идеальными для таких задач, как работа датчиками дождя, температур-

ными датчиками и иными устройствами сбора данных, ставших возможными благодаря беспроводной функции Pi Zero W. А если с устройством вдруг что-то приключится, то заменить его будет дешево и быстро. Короче, это победитель по всем статьям!

➤ Pi W во всей своей красе.



```
sudo apt-get install php-mbstring -y
```

После обновления установка должна работать нормально.

В *Dokuwiki* замечательна крайняя простота в использовании. Здесь нет баз данных или сложных подсистем для настройки. Установка *wiki* сама по себе — это всего одна страница! На странице установки дайте *wiki* соответствующее имя. Вам также понадобится настроить на этой странице суперпользователя. Заполните имя пользователя (нечто короткое и удобное), ваше реальное имя, электронную почту и пароль. Следует также настроить политику ACL. Для домашней сети лучшим планом, вероятно, будет использование закрытой политики с поименованными пользователями. И наконец, если нет особой нужды, установите лицензию на «Не показывать информацию о лицензии». Затем нажмите **Save** [Сохранить]. Процесс установки теперь завершен, так что нажмите

на гиперссылку “Continue to Wiki [Продолжить с Wiki]”. У вас должно получиться войти в систему с помощью указанной вами мгновенно назад аутентификационной информации.

По окончании установки мы рекомендуем администратору войти в Pi и удалить **install.php**. Если вы оставите его на месте, он позволит людям перенастраивать систему. Сделайте это, войдя через SSH в Pi как пользователь “pi”:

```
ssh pi@192.168.0.10
```

Затем удалите файл установки, командой

```
rm /var/www/dokuwiki/install.php
```

### Редактируем и управляем страницами

Вся *wiki* являет собой упрощенную форму языка текстовой разметки. Она использует не HTML, а своего рода разметку соответствия, простую в использовании. Главным заголовкам предшествует шесть знаков равенства (====), и заканчиваются они так же. Создание ссылок на другие страницы сводится к применению стилей языка разметки. Чтобы сделать ссылку на страницу, дайте ей имя и заключите его в двойные квадратные скобки. Например, `[[mygreatsetupage]]`. Теперь страница становится частью *wiki*. По сути, незачем создавать страницу вручную.

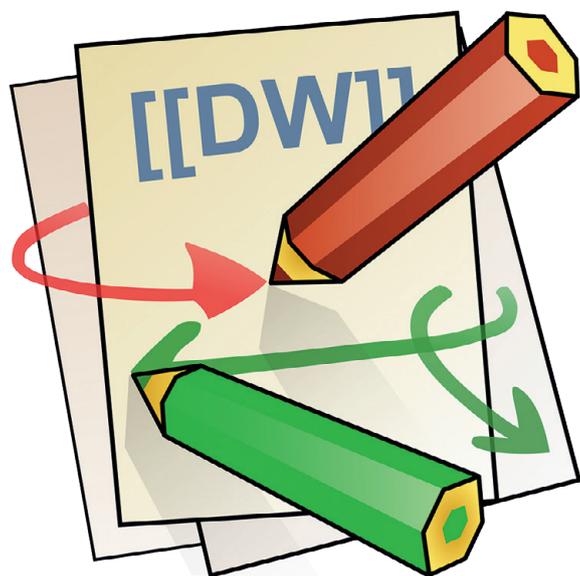
Итак, установка завершена. Хотя это всего лишь простой пример, чего можно добиться с помощью Pi Zero W, он показывает, насколько всесторонним Pi может быть. Особенно привлекательно здесь то, что для создания *wiki* вам не надо создавать виртуальную машину или модифицировать свой локальный компьютер. Эта установка *wiki* была чисто для примера, показывающего, на что способны эти фантастические маленькие устройства.

Но надо сделать предупреждение. Если устройство переместилось в новое место, важно правильно сделать выключение. Судя по всему, Pi куда более чувствителен к отключению питания, чем большинство других систем. Поэтому позаботьтесь, чтобы Pi выключался правильно, с помощью операции `sudo shutdown -h`, тогда это будет чисто.

И напоследок, позаботьтесь сделать резервную копию *wiki* и сохранить ее где-то в безопасном месте. Отличные рекомендации по этому вопросу есть на [www.dokuwiki.org/faq:backup](http://www.dokuwiki.org/faq:backup). **LXF**

### Скорая помощь

Для тех из вас, кто не жаждет командной строки, многие часто используемые опции настройки доступны через инструмент *raspi-config*, включая обновление Pi и настройку опций и пунктов. Только не забудьте перед *raspi-config* поставить `sudo`, чтобы запустить команду с правами root.



➤ Всё стало возможным благодаря *Dokuwiki* с открытым кодом!



# LXF Hot Picks



**Александр Толстой**

нашел пару свободных часов после купания в соленом Черном море, чтобы познакомить вас с горячими летними приложениями с открытым кодом.

**G'MIC-Qt » Ubunsys » HTTraQt » Ring » GsmartControl » Scribus » Krita » KWave » SmartDeblur » Butterfly Effect » Frogatto**

## Обработчик изображений

# G'MIC-Qt

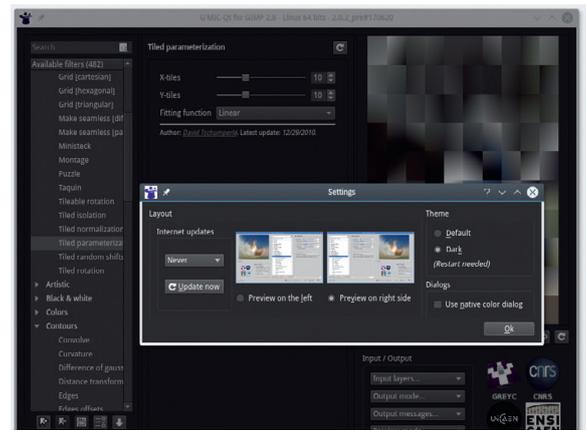
Версия: 2.0 Сайт: <https://github.com/c-koi/gmic-qt>

**В**ыход этого основного релиза *G'MIC* стал долгожданным известием для всех, кто увлекается редактированием изображений в Linux. Основной целевой группой, что вполне очевидно, являются пользователи *GIMP*, потому что теперь *GREYC's Magic for Image Computing* тесно интегрирован с этим знаменитым растровым редактором.

*G'MIC* — полнофункциональная среда с открытым кодом для обработки изображений, включая конвертирование, применение фильтров, манипуляции растровых наборов данных и т. д. Вместе с плагином *gmic\_gimp* проект предлагает утилиту командной строки *gmic* (которая работает

подобно пакету *ImageMagick*), библиотеку C++ под названием *libgmic* и ряд других инструментов. Версия 2.0 *G'MIC* перешла на *Qt5*, хотя до сих пор привязана к *GIMP*, и ее можно найти в меню *Filters* [Фильтры] *GIMP*. *G'MIC-Qt* идеален для фотографов и художников-графиков. Приложение предлагает впечатляюще большое дерево всех возможных фильтров и эффектов, и позволяет поработать с интенсивностью

## Smart Coloring окрасит серию эскизов автоматически.



» Внешний вид интерфейса *G'MIC* легко изменить.

каждого эффекта и увидеть, как будет выглядеть эффект с помощью ползунка *Before* [Перед] и *After* [После].

Выбрав что-то в основном дереве, вы можете настроить такие параметры эффекта, как плотность, радиус или порог. *G'MIC* умеет создавать художественные эффекты, такие как имитация гравюры или чеканки, эскизов или оттисков. И это не всё: *G'MIC 2.0* не был бы столь потрясающим без таких функций, как *Smart Coloring* [Умное окрашивание], способной окрасить серию нарисованных от руки эскизов автоматически и применить текстуры и плавный цветовой переход буквально на лету.

Еще одним приятным новшеством стал инструмент *Color Lookup Tables (CLUT)*, идеальный спутник редакторов изображений RAW. Перетаскивание узлов в куб *CLUT* позволит получать чистые и яркие цвета для ваших фотографий.

В этой новой версии *G'MIC* масса других эффектов — 480, по последним подсчетам — и мы настоятельно рекомендуем вам изучить их самостоятельно. Помните, что *G'MIC-Qt* не является отдельным приложением, поэтому в среднесрочной перспективе можно рассчитывать на интеграцию *G'MIC* с *Krita*, основанным на *Qt* соперником *GIMP*, который в большей мере сфокусирован на цифровом искусстве, чем на обработке изображений. А пока для компиляции программы нужен только *GIMP* и файлы заголовков *Qt*.

## Исследуем интерфейс G'MIC

### Область предпросмотра

Прежде чем применять выбранный фильтр к своему изображению, оцените ожидаемый результат.

### Поиск эффектов

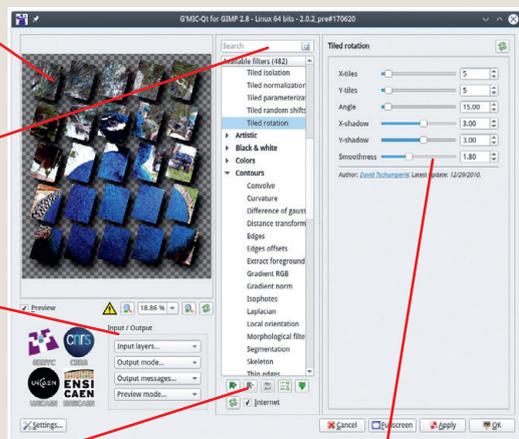
Если дерево ресурсов слишком велико для просмотра вручную, используйте поле поиска вверху.

### Настройка опций ввода/вывода

Используйте эти выпадающие меню для выбора уровней ввода, режима вывода, сообщений и режима предпросмотра.

### Добавочные кнопки поиска

Отмечайте эффекты как *Favourites* [Избранные], переключите дерево в режим *Selection* [Выборка] или покажите весь список эффектов.



### Настройте его по-своему

Каждый из эффектов настраиваемый, и вы сможете легко отменять или активировать эффекты.

## Инструмент администрирования

# Ubunsys

Версия: 2017.06.13 Сайт: <http://bit.ly/2uxx7ji>

**В**округ Ubuntu много шума подняли неутраченные обсуждения перехода с Unity на Gnome Shell. Возможно, для тех, кто не использует дистрибутив семейства Ubuntu, это излишество, но должны же мы отреагировать на запросы основной группы пользователей Linux, которым Ubuntu — дом родной.

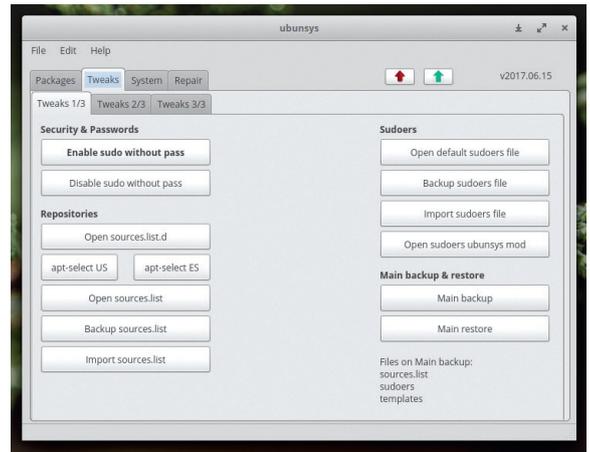
*Ubunsys* — типичное приложение, возникшее как следствие дружелюбия Ubuntu к начинающим пользователям. Многие обычные инструменты администрирования Linux и команды новичков пугают, и им нравятся инструменты вроде *Ubunsys*, да и мы сочли его удобным. Приложение является клиентом на базе *Qt*, который объединяет функции менеджера пакетов и инструмента настройки.

Под строкой меню находятся четыре вкладки под названием Packages [Пакеты], Tweaks [Настройки], System [Система] и Repair [Восстановление]. Раздел Packages не представляет особого интереса — большинство выберет надежный инструмент

*Synaptic*. Однако разделы Tweaks, System и Repair просто отличные, так как они спасают вас от мороки с командной строкой, когда вы не уверены, что делать. Tweaks охватывает основные вопросы, встречающиеся на форумах Ubuntu и в относящихся к Ubuntu разделах на Stack Overflow и Reddit, такие как настройка *Sudo* для работы без пароля, включение режима гибернации, включение брандмауэра или добавление сторонних репозитивов. Ключевое свойство *Ubunsys* — возможность отмены практически любого выполненного вами действия. Большинство действий снабжены кнопкой Disable [Отключить].

*Ubunsys* также работает с обновлениями, и это сбивает с толку многих неопытных пользователей. Программа способна

**Объединяет менеджер пакетов и инструмент настройки.**



» Настроить *sudo* без доступа к терминалу? Да молча!

настроить «умные обновления» для удаления древних ядер, помощи в обновлении до самой последней стабильной версии Ubuntu и т.д.

Как и предполагает название, *Ubunsys* полезен только в Ubuntu и относящихся к нему дистрибутивах, таких как elementary-OS, и привязан к настройкам Ubuntu и таких программ, как *Apt*. Регулярные обновления обеспечивают быструю работу *Ubunsys*, и вам не надо ничего компилировать благодаря свежим Deb-пакетам в разделе Releases на странице проекта на Github.

## Загрузчик сайтов

# HTTraQt

Версия: 1.4.9 Сайт: <http://httraqt.sourceforge.net>

**П**риложение Linux, которое что-то скачивает, обычно ассоциируется с интерфейсом *Wget* или *Aria2*. А вот *HTTraQt* является клоном *WinHTTrack*, инструмента для скачивания сайтов для просмотра офлайн.

Графическое приложение *HTTraQt* и библиотека *HTTrack C++* — это два различных проекта с открытым кодом, разработанные разными командами; но чтобы использовать первое, вам надо сначала установить второе.

Один из обычных вопросов — «Почему ничего не скачивается?». Дело в том, что по умолчанию библиотека *HTTrack* только обрабатывает ссылки внутри данного доменного имени. Она не следует по ссылке, ведущей на другие сайты, не то программа попыталась бы скачать весь Интернет!

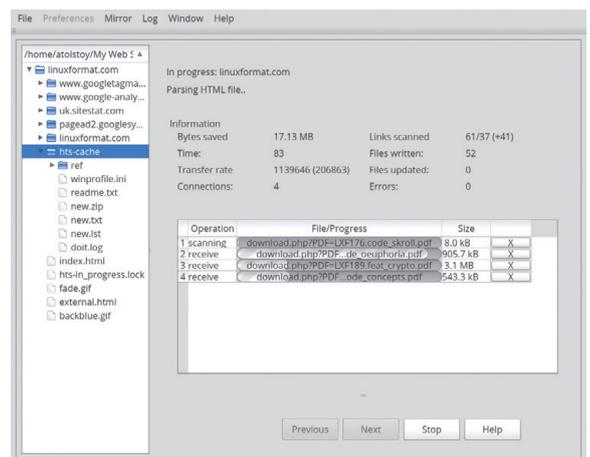
Левая часть окна показывает список ранее скачанных сайтов. Остальная часть панели показывает состояние скачивания и подробности по выделенному в данный

момент пункту. Каждый сайт скачивается в виде ZIP-пакета, содержащего HTML, CSS, JavaScript и все остальные файлы, которые будут работать локально.

Чтобы приступить к работе с *HTTraQt*, запустите его и создайте новый проект для сайта, который хотите скачать. Для каждого проекта вы можете добавить один или несколько URL, обеспечить аутентификацию, настроить прокси-серверы, использовать специальные символы для удаления определенных файлов из зачатки и т.д.

Возможно, самый важный раздел настройки — это структура сайта. Она определяет разные детали, касающиеся полученного в результате локального пакета. Если сайт, который вы пытаетесь скачать,

**HTTraQt выполняет такую же работу, что и Wget, но он гибче.**



» Для тестового прогона мы выбрали самую очевидную цель...

содержит большое количество сложных JavaScript или иной динамический код, имеет смысл перед началом скачивания всё это как следует настроить.

*HTTraQt* выполняет такую же работу, что и *Wget* (с должными опциями командной строки), но он намного более полнофункциональный — и гибче. Вы можете изменить данные аутентификации, настроить разное время ожидания, детали и многие другие экспертные опции.

Платформа коммуникации

# Ring

Версия: 2 beta Сайт: <http://bit.ly/2txm7Cr>

**Н**едavno Microsoft прекратила поддержку родного клиента Skype в Linux, на базе Qt, и заставила всех перейти на более новый, но очень ограниченный и ресурсоемкий Skype, основанный на Electron. В сущности, это немного измененная внешне и слегка настроенная версия web-версия Skype.

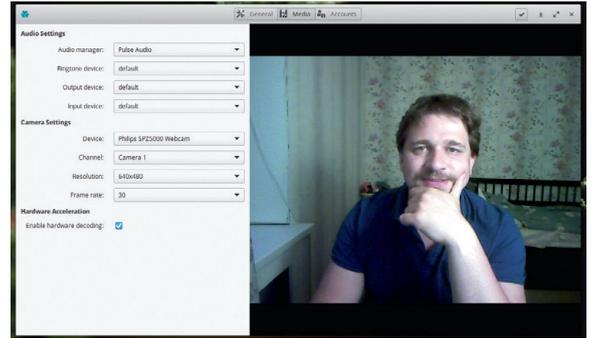
Это привело к появлению спроса на альтернативное коммуникационное ПО, возможно, и с открытым кодом, способное работать с безопасным обменом сообщениями и видео звонками. Мы полагаем, что Ring может стать достойной заменой, и вот почему.

GNU Ring — это голосовое ПО IP, одновременно и дружелюбное к пользователю, и профессиональное по природе. Оно поддерживает аудио/видеокоммуникации с переадресацией звонка, удержанием и автоматической настройкой качества аудио... всё как в классической версии Skype до покупки его Microsoft. Ring поддерживает

децентрализованные (одноранговые) звонки, но также добавляет поддержку Asterisk благодаря возможности использовать учетные записи SIP.

Программа состоит из двух основных частей: демона и клиента. Демон выполняет все логические действия в фоновом режиме, а клиент на самом деле является GUI, с которым вы взаимодействуете. Есть множество клиентов для Ring, но достаточно будет упомянуть дружелюбный к Plasma клиент на <https://github.com/KDE/ring-kde> и набор клиентов, поддерживаемых напрямую Savoir-faire Linux, компанией, занимающейся технологией Ring. Это официальные Ring Gnome на GTK3 и Ring for Android (найдите его на Google Play).

## Поддерживает коммуникации с переадресацией звонка.



➤ Проверьте настройки, прежде чем делать видеозвонок — и постарайтесь выглядеть не хуже!

При первом запуске клиента (каким бы он ни был), он попросит вас создать учетную запись, и именно в этот момент вы выбираете ник и пароль. Затем можно приступить к обмену сообщениями со всеми, чьи ники вы знаете.

Простейшим способом протестировать Ring будет создание учетной записи на своем компьютере Linux и еще одной — на своем смартфоне, и попытка пообщаться с самим собой. Желая настроить собственный сервер Ring, загляните на <https://github.com/weavejester/ring-server>. Это позволит предусмотреть «запертый сад» для коммуникации, если вам страшно довериться открытому серверу Ring.

Диагностика диска

# GSmartControl

Версия: 1.0.1 Сайт: <http://bit.ly/2tZbJWU>

**Д**олгие годы все жесткие диски и SSD поддерживали SMART (self-monitoring, analysis and reporting technology — технологию самопроверки, анализа и отчета), которая заранее предупредит вас о скорой кончине жесткого диска, чтобы вы успели перенести свои данные.

Linux всегда поддерживал SMART, однако частенько возникала неопределенность насчет того, как именно получать эти предупреждения накануне отказа диска. Раньше вы могли использовать такие команды, как smartctl или smartd из пакета Smartmontools, или выбрать чудесное приложение Gnome Disks, предлагающее шикарный GUI для smartctl.

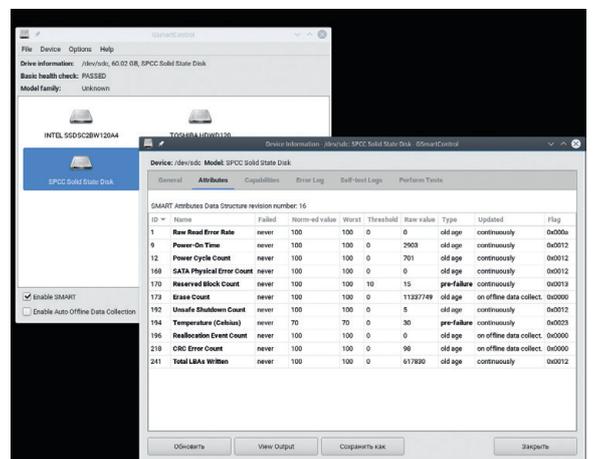
Но если ваш рабочий стол не Gnome и вы при этом хотите получить доступ к smartctl через интерфейс, подумайте о GSmartControl. Это отдельное приложение, которое недавно было портировано в GTK3 и доступно для Linux, Windows и Mac OS.

GSmartControl идеально для пользователей, не особо сильных в технологиях,

но достаточно сведущих в компьютерной технике, чтобы понимать необходимость поддержания жесткого диска в хорошей форме. Приложение показывает значки всех физических дисков в вашей системе, как менеджер файлов. Вы можете быстро выяснить точный номер модели каждого устройства и посмотреть, прошло ли оно базовую проверку работоспособности. Просто дважды щелкните по значку, чтобы открыть окно со всеми подробностями, которые Smartctl обычно выводит в терминал.

GSmartControl делит всё на шесть вкладок и позволяет изучить функции вашего жесткого диска, просмотреть журналы ошибок, отобразить время жизни диска в часах и т.п. Одна из вкладок позволяет вам вручную проводить некоторые тесты

## Эта программа позволяет провести тесты на заданном диске.



➤ У этого старого SSD закончились резервные блоки. Можно смело прогнать, что не сегодня-завтра у него будут проблемы.

на заданном диске. Вы можете выбрать короткую, расширенную или опосредованную самопроверку и увидеть, действительно ли ваш жесткий диск соответствует тому, что о нем говорит программа.

Нам понравился GSmartControl за его полезность и компактный размер, который позволит применять его на минималистских рабочих столах без привлечения большого количества нежелательных зависимостей. Разработчик предлагает много пакетов для всех основных дистрибутивов Linux, поэтому для большинства читателей установка GSmartControl сведется к нескольким щелчкам мышью.

## Настольная издательская система

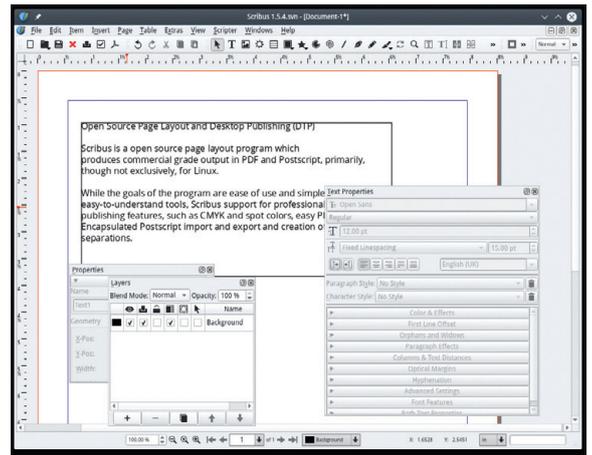
## Scribus

Версия: 1.5.3 Сайт: [www.scribus.net](http://www.scribus.net)

**М**ы неоднократно рассказывали о *Scribus* на страницах *Linux Format*. Отчасти причина в том, что это приложение рабочего стола для публикаций является прямым соперником *InDesign* от Adobe, которым команда **LXF** ежедневно пользуется для создания журнала, который вы сейчас держите в руках. Программа существует с 2003 г. и сейчас многими считается достойным и качественным примером приложения с открытым кодом. Она весьма замысловатая и многофункциональная, и при этом ее достаточно легко освоить. И если вам непонятно, чего ради выбирать *Scribus* вместо, скажем, *LibreOffice Writer*, то читайте дальше.

*Scribus* — идеальное решение для создания многостраничных публикаций, например, журналов, брошюр и книг, печатаемых по новейшей технологии прямого экспонирования пластин на общепринятом принтере. Вы можете разместить объекты на странице в рамках-фреймах и заставить

текст автоматически перетекать с одной страницы на другую, управлять цветовыми профилями ICC и экспортировать свой дизайн как PDF-файл профессионального качества. По сути, *Scribus* объединяет инструменты векторного и текстового редакторов, затем вводит управление цветом и прочие специальные функции. Одним из слабых мест *Scribus* всегда была низкая производительность при работе со сложным дизайном. Как только у вас появляется несколько страниц, густо заполненных текстом и встраиваемыми объектами, *Scribus* начинает ужасно тормозить. Однако похоже, что проблемы с производительностью оставлены в прошлом: серия 1.5.x



» Самое главное — это улучшенная производительность *Scribus*, которая больше всего нас впечатлила.

*Scribus* наконец-то соответствует профессиональным продуктам DTP. Движок верстки текста, ее рендеринг и предпросмотр, копирование и вставка принялись работать без заминок. Сейчас в *Scribus* появились долгожданные улучшения, например, предпросмотр шрифтов, поддержка импорта KRA-файлов Krita и хранение XDG-совместимых настроек. Более того, после автоматической проверки кода исправлено много ошибок. Если вы занимаетесь публикациями на ПК, то последний релиз *Scribus* просто непременно должен у вас быть, и пользоваться им — истинное удовольствие.

**Идеальное решение для дизайна журналов, брошюр и книг.**

## Редактор изображений

## Krita

Версия: 4.0.0 pre-alpha Сайт: <https://krita.org>

**Е**сли задать вопрос о лучшей графической программе, многие будут отлично отзываться о *GIMP*, сопернике Adobe *Photoshop* с открытым кодом. Однако это не единственная художественная программа в мире Linux...

*Krita* — многофункциональный профессиональный пакет для цифровых художников. В прошлом году мы рассказывали о версии 3.0.1 (**LXF217**), но в мире открытого кода изменения — дело обычное и частое, и разработчики *Krita* недавно предложили революционную *Krita 4.0 Alpha*. Как вы и заподозрили, это не окончательная версия, а скорее витрина множества новых и потрясающих функций, запланированных для этого приложения.

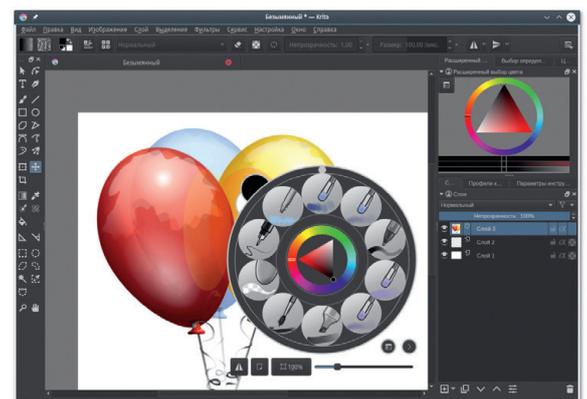
Данная версия предлагает совершенно новую векторную систему слоев; в ней для сохранения векторной информации теперь используется формат SVG, вместо Open Document Graphics (ODG). А значит, вы можете начать создавать произведения, скажем, в *Inkscape* и затем перенести

редактирование в *Krita*. Вы даже можете перетащить SVG-файлы в документ *Krita*, и они вставятся в виде векторных слоев. Помните, что *Krita 4* может открывать файлы, созданные в версии 3, но не наоборот.

Далее, пересмотрена система Airbrush, позволяющая добиться лучших результатов, новая кисть Healing и инструменты Text, пересмотрен докер палитры, позволяющий группировать цвета и редактировать образцы, просто нажав на них. Если в прошлом вам нравилось работать в *Krita*, вы увидите, что сейчас изменений больше, чем при обновлении программы с 2.x до 3.0.

*Krita* по-прежнему идеальный инструмент для создания комиксов и даже простой анимации, поскольку предлагает богатый спектр соответствующих шаблонов

**Данная версия предлагает новую векторную систему слоев.**



» Как и *Photoshop*, *Krita* умеет работать и с растровыми, и с векторными объектами — причем без приклеенного ценника.

при выборе File > New [Файл > Новый]. Хотя сильная сторона *Krita* заключается в создании рисунков методом рисования, программу можно использовать и для ретуширования фотографий или иных изображений. Готовый к использованию ArtImage доступен на сайте проекта, поэтому вам незначительно беспокоиться по поводу поиска самого последнего пакета для вашего дистрибутива Linux или самостоятельно компилировать код. Отличная работа, а *Krita* — действительно отличный проект!

## HotGames Развлекательные приложения

## Физическая игра

# Butterfly Effect

Версия: 0.9.3 Сайт: <http://the-butterfly-effect.org>

**З**акон физики нам не изменить, но мы можем изучить их и понаблюдать за разными эффектами, вызванными гравитацией, ветром, движением качения и т. д. *Butterfly Effect* — правдоподобная физическая имитация в виде медитативной головоломки, разделенной на уровни.

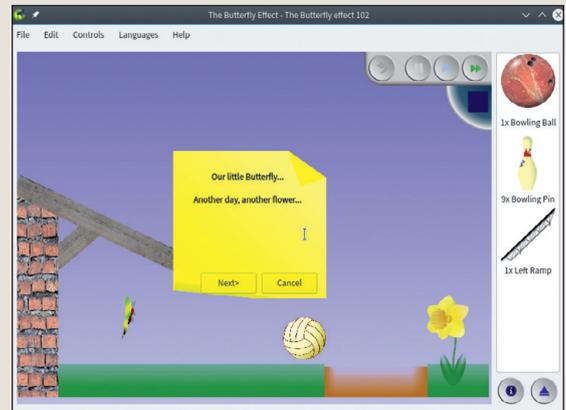
*Butterfly Effect* [Эффект Бабочки] пока что, по мнению ее разработчика, находится в стадии альфа, однако в нее вполне можно играть. Название у игры такое же, как у фильма о путешествии во времени, в соответствии с чем нам пришлось погоняться за пыльным экземпляром LXF136, где мы в последний раз о ней рассказывали. С тех пор произошли некоторые изменения. Возможно, самое примечательное — количество уровней: теперь их 76!

Каждый уровень игры — головоломка, которую вы должны решить. Вам надо

иметь дело с разными объектами: например, кеглей для боулинга, волейбольным мячом или воздушным шариком, которые будут падать, ускоряться на наклонной поверхности и взаимодействовать друг с другом. На правой части экрана размещены объекты, которые можно добавлять на игровое поле.

Уровни, как правило, хорошо продуманы и дают вам свободу решать каждую головоломку одним из нескольких способов. Уровень сложности возрастает по мере вашего продвижения в игре, но даже когда вы упираетесь в кирпичную стену и вам не удается найти решение, *Butterfly*

**Вам надо иметь дело с разными объектами: кеглей или мячом.**



» Всё просто: дайте бабочке долететь до цветка.

*Effect* дает возможность заново проходить каждый уровень сколько угодно раз.

В целом игра является хорошим тренажером для мозга и достойным инструментом для обучения детей основным физическим концепциям. Автор намеревается выпустить версию 1.0, как только отладит все уровни, из которых 63 идеально пригодны для игры уже сейчас. Поэтому не сомневайтесь, скачайте себе игру для вашего дистрибутива с сайта проекта.

## Аркада

# Frogatto

Версия: GIT Сайт: <http://bit.ly/2tdsRUe>

**И**ногда встречаются игры, о которых мы бы хотели написать в *HotPicks*, но их лицензионная модель скрывает исходный код. Обычно у подобных игр с закрытым кодом очень богатая по цвету графика, высококачественная музыка и очень захватывающий сюжет, и всё это служит одной цели — завлечь вас и заставить оплатить полную коммерческую версию или приобрести некие объекты в самой игре за реальные деньги.

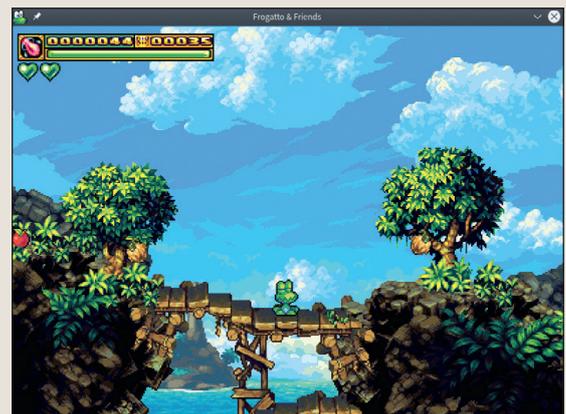
Помня обо всем этом, мы почти пропустили *Frogatto*, игру-платформер, которая когда-то продавалась в Humble Store примерно за \$10. К счастью, игра оказалась с открытым кодом, с 558 МБ фрагментов кода, доступными на Github. Вы управляете лягушкой, которая исследует огромный внешний мир, побеждает своих врагов, находит секретные объекты и собирает монеты, чтобы потратить

их в игре. На первый взгляд, ничего особенного в этой аркаде нет, но нас впечатлили вложенные в нее труды и усилия.

Основных режимов два: casual [обычный] и challenging [трудный]. Оба начинаются в доме лягушки, где мы выясняем, что наш герой проспал и ему надо идти искать работу.

Помимо прекрасной и подробной прорисовки, *Frogatto* предлагает захватывающую механику игры и мягкий способ устранения врагов. Лягушка ловит их языком, после чего (опционально) выплевывает. Прыжки через беспомощных врагов заставляют тех волшебным образом исчезнуть, за что добавляются очки.

**Прыжки через врагов заставляют тех исчезнуть.**



» Лягушке очень нравится пейзаж без злобных муравьев...

В игре есть новаторские контрольные точки в виде уличных туалетных кабинок и уровень дружеского руководства, который ознакомит вас с игровым миром. *Frogatto* также предлагает несколько игровых опций. Это нечто вроде встроенных мини-игр, независимых от основной игры. В общем, мы бы сказали, что *Frogatto* — увлекательная игра, которая поможет вам скоротать время.

## Звуковой редактор

## KWave

Версия: 17.04.2 Сайт: <https://github.com/KDE/kwave>

**K**Wave всегда был в тени *Audacity*, который является, наверное, самым передовым аудиоредактором в Linux. Вас, возможно, удивит, что *KWave* существует с 1998 г., и это небольшое стабильное приложение весьма удобно для определенных задач.

Например, вы хотите отрезать тишину в начале трека, увеличить, уменьшить или нормализовать громкость, добавить эффект затухания/усиления звука или просто сохранить аудиофайл в другом формате. Для подобных простых задач вам не нужно тяжеловесное приложение — *KWave* будет более чем достаточно.

По умолчанию любому новому файлу в *KWave* отведено два трека, но можно добавить еще — иными словами, сделать наложение разных музыкальных частей и мастеринг многотрековой композиции. Визуальная волна проста в использовании и позволяет выделять части, изменять масштаб, копировать и вставлять, и т. д. Естественно, *KWave* умеет записывать звук,

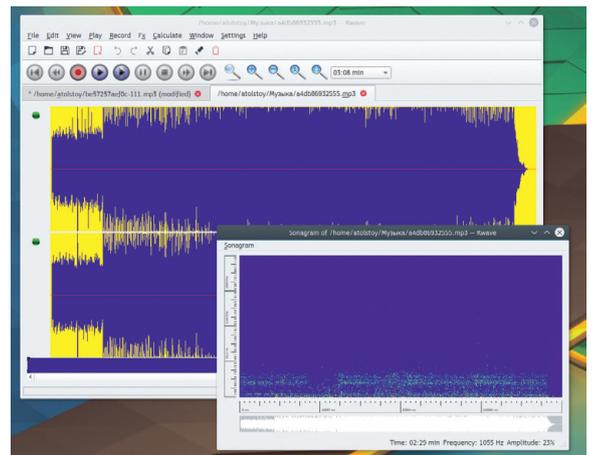
при наличии микрофона, поэтому вы можете легко выразить свою креативность.

Для ввода и вывода аудио в установке *KWave* по умолчанию используется среда Qt Multimedia, но можно настроить *PulseAudio*, ALSA или даже устаревшую OSS через окно *KWave Settings* [Настройки]. Редактор не может похвастаться изобилием эффектов, но частично это оправдывается умением программы удаленно применять изменения. Например, требуемая запись быстро загрузится командой

```
$ kwave --iconic --disable-splashscreen sample.wav
```

Кроме того, *KWave* поддерживает удаленное управление — через такие методы, как `kwave:plugin %3Aexecute?normalize, kwave:save`.

**Визуальная волна проста и позволяет выделять части.**



► Просматривайте элементы компрессии MP3 с помощью Sonogram.

В сочетании с *Eko*, еще одним минималистским звуковым редактором (см. **LXF214**), *KWave* может стать настоящим сердцем вашей собственной студии звукозаписи. Запишите свой голос, создавайте подкасты, преобразуйте свои WAV-файлы в MP3, добавляйте инструментальный раздел — и станьте, наконец, поп-звездой! Редактор является частью пакета KDE Applications, и его политика релизов синхронизована с остальными приложениями в подборке.

## Усилитель яркости изображения

## SmartDeblur

Версия: 1.27 Сайт: <http://bit.ly/2ui7PZx>

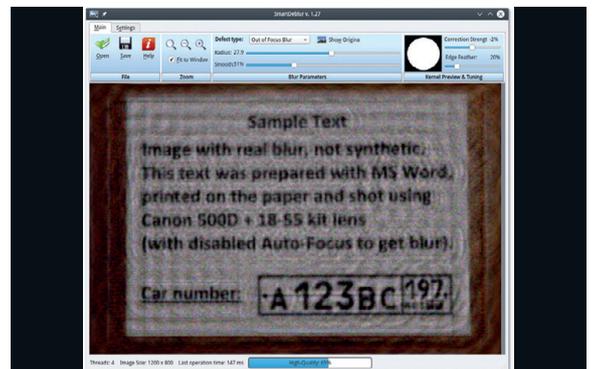
**В**ы можете предположить, что *HotPicks* — это всегда свежес выпущенные или обновленные программы, и по большей части будете правы. Одним исключением будет *SmartDeblur*. Этот любительский проект по-прежнему посживает на GitHub в качестве программы с открытым кодом, хотя он не обновлялся уже несколько лет, в то время как более поздние версии стали проприетарными и их можно приобрести на <http://smartdeblur.net>.

*SmartDeblur* — программа восстановления фотографий, которая исправляет размытость при движении, плохую фокусировку и размытие по Гауссу. Большинство инструментов для редактирования фото сначала надо купить, поэтому оплату \$49 за текущую версию *SmartDeblur* можно считать разумным ходом; но мы так не делаем. Мы лучше воспользуемся предыдущей версией 1.27, программой с открытым кодом в чистом виде — к проприетарным программам обращаться незачем!

*SmartDeblur* — это графический интерфейс технологии восстановления фокусировки, которая изменяет соседние пиксели с помощью матрицы свертки. Единственный инструмент, способный делать то же самое — плагин *Refocus* для *GIMP*, еще более древняя программа, которую вы вообще не встретите в наши дни без прямой ссылки (<http://bit.ly/2tN8wGY>).

*SmartDeblur* намного лучше и удобнее в использовании, потому что загружаемое изображение автоматически обновляется, когда вы перетаскиваете ползунки приложения и пытаетесь тонко настроить эффекты. При первом запуске *SmartDeblur* показывает образец изображения с размытым текстом и предлагает вам сделать его

**Это графический интерфейс восстановления фокусировки.**



► Если вы что-то здесь прочли, значит, приложение работает!

читаемым. Это наилучший способ научиться балансировать *Radius* [Радиус], *Smoothness* [Гладкость] и *Correction Strength* [Предел Исправления]. На обработку больших изображений *SmartDeblur* потребует гораздо больше времени, и, возможно, имеет смысл заранее обрезать (не меняя размер!) свое изображение и применить правильные настройки к копии, а потом уж работать с файлом оригинала. Приложение поможет вернуть к жизни фотографии не в фокусе, добавить резкости к сканам старых изображений и усилить яркость изображений с низким разрешением. Для успешной компиляции программе нужны *Qt4* и заголовки *FFTW*, хотя некоторые дистрибутивы Linux уже имеют ее в своих репозиториях. **LXF**

# На диске

Дистрибутивы, приложения, игры, книги и всякое-разное...

Лучшее из Интернета, упакованное в 8 ГБ качественного DVD.



## Дистрибутивы

Обычно наши DVD содержат три-четыре дистрибутива на каждой стороне, хотя мы умудрялись втиснуть шесть отдельных дистрибутивов на одну сторону с помощью кувалды! Исторически нормой было четыре, но теперь у нас обычно есть место не более чем три на сторону. Причин увеличения размера дистрибутивов может быть две. Содержащиеся в них программы — а сюда входят приложения рабочего стола и основная ОС — могут разрастись. Или их становится больше по количеству. И что: свободное ПО становится объемнее или его просто становится больше?

Как обычно с такими вопросами, ответ находится где-то посередине. Стало намного больше свободных программ, но они еще и обрастают функциями. Уместность раздутости программ зависит от того, будете ли вы применять эти функции, и даже от того, знаете ли вы о них. Лично я скорее выбрал бы программу с десятью функциями, которые мне никогда не пригодятся, чем пропустил функцию, которая мне действительно нужна. В этом и смысл программ для всех. Нужды у пользователей разные, поэтому стремление к универсальности может привести к увеличению объема.

Теперь всё, что нам требуется — это 6-ГБ DVD...

Neil

## » Важно ВНИМАНИЕ!

### Порченые диски

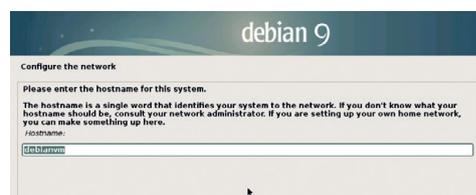
В маловероятном случае какого-то дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу [disks@linuxformat.ru](mailto:disks@linuxformat.ru) или телефону +7 (812) 309-0686.

Самый свежий релиз

# Debian 9.0.1

Когда речь идет о совершенстве и стабильности, Debian часто считается золотым стандартом. И этот релиз не должен быть исключением; однако в программе установки есть несколько аспектов, о которых вы должны знать. Первый — у вас нет опции запуска программы установки с рабочего стола live. Вместо этого надо загрузиться в программу установки, и это не работает при загрузке прямо с *Linux Format* DVD. Это не критика Debian: он никогда и не предназначался для работы подобным образом. Решение простое. Запишите ISO Debian с *Linux Format* DVD на пустой DVD или USB-брелок и загрузитесь оттуда, именно это и предполагалось разработчиками Debian. Затем у вас будет две опции установки: одна с помощью интерфейса с окнами и базовая текстовая альтернатива.

Другая проблема вас может вообще не затронуть, но если она вас всё же коснется, вы об этом узнаете! Процесс установки не будет успешным без подключения к Интернету при его запуске. Если вы используете беспроводное соединение, у вас есть опция ввести пароль Wi-Fi, чтобы настроить соединение, но если соединения не будет, установщик не будет работать. Вы можете пропустить настройку сети, но программа установки зависит от получения информации о релизе от одного из серверов Debian, и выдаст сбой, если не сможет с ним соединиться. Итак, не пытайтесь установить Debian Stretch без соединения с сетью. Вы увидите, что это версия 9.0.1. У версии 9.0 тоже имелись



» К сожалению, вам понадобится рабочее подключение к интернет, чтобы запустить текущую программу установки Debian.

проблемы с установщиком (был включен некорректный файл), и ее надо было быстро обновить.

Это не очень соответствует легендарным стандартам тестирования Debian продолжать тесты, пока всё не заработает идеально, а потом еще немного потестировать. Однако проблемы есть только у программы установки, и как только Debian окажется на вашем жестком диске, вы получите всё, чего можно от него ожидать: дистрибутив, у которого, может быть, и не самая свежая версия абсолютно всего, но дистрибутив, надежный, как скала. Мы включили Gnome-версию live-релиза Debian, хотя у Debian нет рабочего стола по умолчанию, и поэтому, как только он будет на вашем компьютере, вы можете добавить рабочий стол несколькими нажатиями в *Synaptic*.

Аутентификационная информация: имя пользователя — **user**, пароль — **live**.



**Новый релиз** **Debian 9.0**  
Исследуйте прочный как скала дистрибутив серверного класса для всех!  
Рабочий стол GNOME Ядро 4.9 LTS Новые средства анализа MariaDB PHP 7.0

**Voyager Live 9.0**  
Дружелюбный дистрибутив Linux с предустановленным рабочим столом  
Ядро 4.9 Рабочий стол Xfce Интерфейс для Dock Виджеты Conky

**LINUX** **LIVE-ДИСК: ГОТОВ К РАБОТЕ**  
ВСЕ НЕОБХОДИМОЕ ДЛЯ СТАРТА В LINUX

# Новичок в Linux? Начните отсюда!

- » Что такое Linux? Как его установить?
- » Есть ли в нем эквивалент MS Office?
- » Зачем нужна командная строка?
- » Как устанавливать программы?

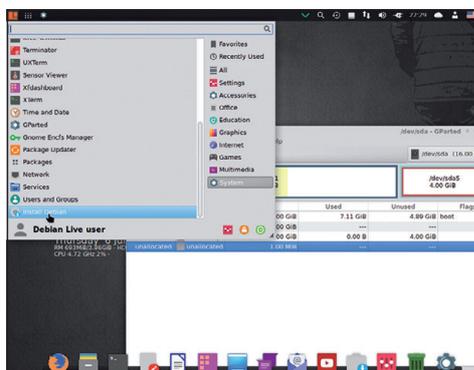
Ответы приводятся в [Index.html](#) на диске.

Дистрибутив на базе Debian

# Voyager Live 9.0

Voyager — французский дистрибутив с дружелюбным интерфейсом и репутацией. Он поставляется в двух редакциях: версия на Ubuntu и та, которая у нас здесь, на базе Debian. Это самый свежий релиз Voyager, основанный на новом Debian 9, но с индивидуальным рабочим столом Xfce. Xfce — относительно легковесный, но при этом полнофункциональный рабочий стол, находящийся где-то посередине между тяжеловесными рабочими столами от KDE и Gnome и очень легковесными, такими как LXDE и IceWM.

Voyager использует программу установки Debian, но в данном случае ее можно запустить из рабочего стола live: вы найдете ее в разделе System [Система] основного меню. Копировать ISO Voyager на диск или USB-брелок не надо. Однако у вас может возникнуть проблема, когда программа установки сообщает, что не может создать файловую систему root. Самый простой способ избежать этого — запустить GParted с рабочего стола live, чтобы разобраться с разбиением на разделы, оставив невыделенное место под Voyager, а потом запустить программу установки. Возможно, вам также удастся обойти эту проблему, перейдя в терминал и убедившись, что ни один из разделов вашего

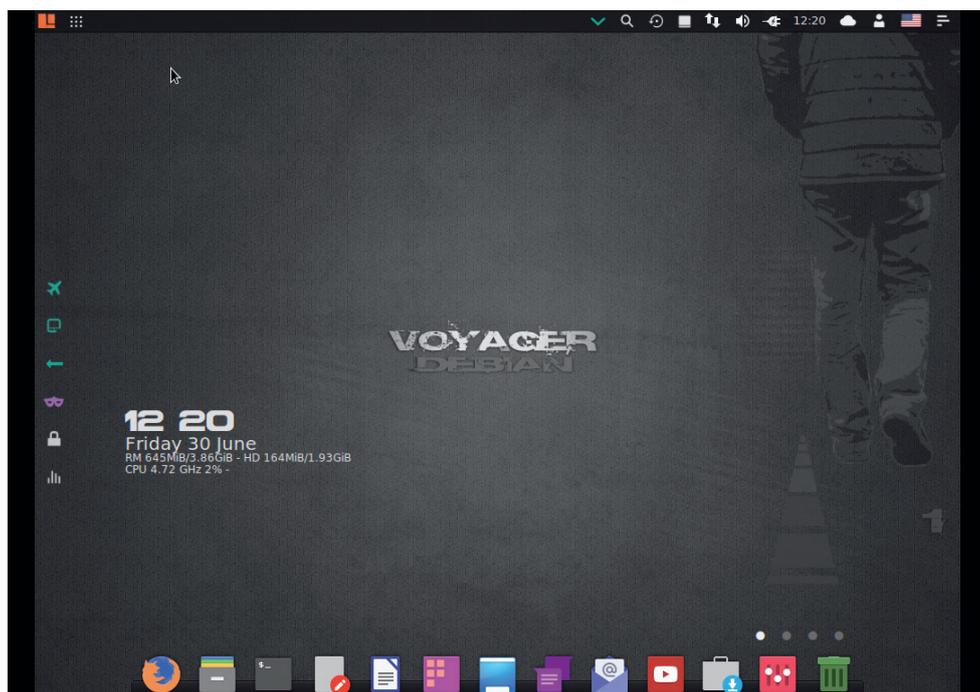


» Безопаснее всего перед запуском установщика Voyager высвободить некоторое место в GParted.

жесткого диска не смонтирован перед шагом разбиения на разделы; однако использование GParted — более здоровое решение.

Хорошая новость в том, что Voyager прекрасно устанавливается на компьютер без подключения к Интернету.

Аутентификационная информация: имя пользователя — `user`, пароль — `live`.



## И еще!

### Системные инструменты

#### Главное

**Checkinstall** Установка tar-архива с помощью менеджера пакетов.

**GNU Core Utils** Основные утилиты, которые обязаны присутствовать в каждой операционной системе.

**Hardinfo** Инструмент тестирования системы.

**Kernel** Исходный код самого последнего стабильного релиза ядра.

**Memtest86+** Проверьте ОЗУ на предмет неисправностей и сбоев.

**Plopp** Простой менеджер загрузки для запуска ОС с CD, DVD и USB.

**RaWrite** Создавайте загрузочные дискиеты в MS-DOS в Windows.

**SBM** Независимый от ОС менеджер загрузки с несложным интерфейсом.

**WvDial** Соединяйтесь с Интернетом через телефонный модем.

### Чтение

#### Книжная полка

**Расширенное руководство по скриптам Bash** Изучите написание скриптов еще лучше.

**Руководство Bash для начинающих** Овладейте написанием скриптов Bash.

**Руководство по созданию скриптов Bourne Shell** Начните осваивать скрипты оболочки.

**The Cathedral and the Bazaar [Собор и Базар]** Классический текст Эрика С. Реймонда [Eric S Raymond], объясняющий преимущества открытой разработки.

**Справочник администратора Debian** Базовое руководство для системных администраторов.

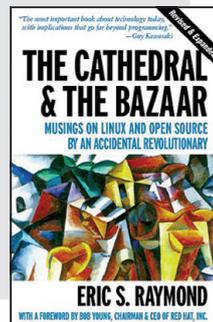
**Введение в Linux** Удобное руководство со множеством подсказок для начинающих пользователей Linux.

**Словарь Linux** Терминология Linux от А до Я.

**Linux Kernel in a Nutshell [Ядро Linux в Двух Словах]** Введение в ядро, написанное великим магистром ядра Греггом Кроа-Хартманом [Greg Kroah-Hartman].

**Руководство системного администратора Linux** Контролируйте свою систему.

**Обзор инструментов** Полный обзор инструментария GNU.



# Пропустили номер?

**LINUX**  
FORMAT  
Главное в мире Linux

Закажите его на сайте [www.linuxformat.ru](http://www.linuxformat.ru) в «ГНУ/Линуксцентре»!  
Журналы доставляются и в печатной, и в электронной форме,  
так что получение нужного вам выпуска LXF может занять всего  
пару минут с момента открытия браузера!

## LXF226 Август 2017



- » Уходим на Linux Выбираем свободу
- » Раскидываем сети Ну-ка, проследи!
- » Анимация Почувствуйте себя Диснеем
- » Меньше ада Doom на шоколадном движке

**LXF DVD:** BackBox Linux, Elementary OS, Manjaro, Tails, Zorin, горячие новинки и прочее...

**Печатная версия:** [shop.linuxformat.ru/lxf\\_226/](http://shop.linuxformat.ru/lxf_226/)  
**PDF-версия:** [shop.linuxformat.ru/elxf\\_226/](http://shop.linuxformat.ru/elxf_226/)

## LXF225 Июль 2017



- » Притворимся хакерами Но не корысти ради
- » Мультки и не только Программы анимации
- » Параллельный Интернет Сеть для избранных
- » Лучше перебить Двухфакторная аутентификация

**LXF DVD:** Ubuntu, Solus, Android-x86, 12 книг о Linux (на английском), горячие новинки, архив LXF за 2005–2016 гг. и прочее...

**Печатная версия:** [shop.linuxformat.ru/lxf\\_225/](http://shop.linuxformat.ru/lxf_225/)  
**PDF-версия:** [shop.linuxformat.ru/elxf\\_225/](http://shop.linuxformat.ru/elxf_225/)

## LXF224 Июнь 2017



- » Конец Unity Ubuntu перешел на Gnome
- » Где KDE краше Лучший дистрибутив с KDE
- » Боронися, бабка Брандмауэр + роутер = защита
- » Кому верить? Надежные ключи шифрования

**LXF DVD:** Ubuntu, Linux Lite, Manjaro, Netrunner, 12 книг о Linux (на английском), горячие новинки и прочее...

**Печатная версия:** [shop.linuxformat.ru/lxf\\_224/](http://shop.linuxformat.ru/lxf_224/)  
**PDF-версия:** [shop.linuxformat.ru/elxf\\_224/](http://shop.linuxformat.ru/elxf_224/)

## LXF223 Май 2017



- » Ставим на Zero Raspberry Pi Zero W уже с Wi-Fi и Bluetooth
- » Дизайн для 3D-принтера Создание моделей
- » В стиле Blu-ray Заставим работать в Linux
- » Стеганография И чтоб никто не догадался

**LXF DVD:** feren OS, IPFire, openSUSE Tumbleweed, RancherOS, Rescatux, Ubuntu Studio, XenialDog, 12 книг о Linux (на английском), горячие новинки и прочее...

**Печатная версия:** [shop.linuxformat.ru/lxf\\_223/](http://shop.linuxformat.ru/lxf_223/)  
**PDF-версия:** [shop.linuxformat.ru/elxf\\_223/](http://shop.linuxformat.ru/elxf_223/)

## LXF222 Апрель 2017



- » Стать Linux-джедаем Изучаем терминал
- » В Интернет — невидимкой Дистрибутивы конфиденциальности
- » Серверы на все лады Два учебника и статья
- » Сайт с галереей Адаптивный web-дизайн

**LXF DVD:** Devuan, Linux Mint, ROSA, Scientific Linux, Ubuntu Budgie, 12 книг о Linux (на английском), горячие новинки и прочее...

**Печатная версия:** [shop.linuxformat.ru/lxf\\_222/](http://shop.linuxformat.ru/lxf_222/)  
**PDF-версия:** [shop.linuxformat.ru/elxf\\_222/](http://shop.linuxformat.ru/elxf_222/)

## LXF221 Март 2017



- » Прощай, Google! Мы найдем получше
- » Ваш добрый демон Выбор ОС BSD
- » Сервер на халяву CentOS, клон Red Hat
- » Linux+Windows Сожительство возможно

**LXF DVD:** ExTiX, Porteus, Rebecca Black Linux, Siduction, Ubuntu, Zorin, 12 книг о Linux (на английском), горячие новинки и прочее...

**Печатная версия:** [shop.linuxformat.ru/lxf\\_221/](http://shop.linuxformat.ru/lxf_221/)  
**PDF-версия:** [shop.linuxformat.ru/elxf\\_221/](http://shop.linuxformat.ru/elxf_221/)

Подпишитесь на печатную версию журнала на [www.linuxformat.ru/subscribe](http://www.linuxformat.ru/subscribe)  
или [www.linuxcenter.ru](http://www.linuxcenter.ru), и получите электронную версию в подарок!

Телефоны отдела подписки

- » Санкт-Петербург: (812) 309-0686
- » Москва: (499) 271-4954



Linux Format ВКонтakte:  
[vk.com/linuxform](https://vk.com/linuxform)

# » Содержание

**Linux**  
FORMAT

## Страница 1

### ДИСТРИБУТИВЫ

**Debian 9.0 Stretch** (64-битный)  
Полностью свободная ОС на базе Linux. Содержит все необходимые для полноценной работы инструменты, поставляется с более чем 50000 пакетов (прекомпилированными ПО в формате, легко устанавливаемом на вашем ПК). Разработчики прилагают все силы для скрупулезного тестирования всех компонентов, поэтому Debian чрезвычайно популярен как серверный дистрибутив.

**Уоуагер Live 9** (64-битный)  
Дистрибутив и liveDVD на базе Xfce, с опенчанным рабочим столом Xfce. Включает Avant Window Navigator или AWN (панель навигации в виде dock-станции), Solix (монитор системы) и более 300 фотографий и анимаций, пригодных в качестве обоев рабочего стола.

### ПОМОЩЬ

Руководство новичка  
Руководства  
Ответы  
Часто (FAQ)

### НОПРИСКС

**Винтерфелт 0.9.3** Игра с физикой  
[the-winterfell-effect.org](http://the-winterfell-effect.org)

**Fogato** (64) Игра-аркада  
[bit.ly/29S8NDe](http://bit.ly/29S8NDe)

**СМПС-01-20** ПО для обработки изображений  
[bit.ly/29DkxWU](http://bit.ly/29DkxWU)

**Семантикол 1.0.1** Диагностика диска  
[bit.ly/29DkxWU](http://bit.ly/29DkxWU)

**НТТрад 1.4.9** Загрузчик сайтов  
[bit.ly/29DkxWU](http://bit.ly/29DkxWU)

**Книга 4.0.0 re-alpha** Редактор изображений  
[bit.ly/29DkxWU](http://bit.ly/29DkxWU)

**Клара 0.0**

**Квиев 17.04.2** Редактор звука  
[bit.ly/29DkxWU](http://bit.ly/29DkxWU)

**onlycode.com/KCE/kyakaze**

**Ring 2 beta** Платформа для общения  
[bit.ly/29DkxWU](http://bit.ly/29DkxWU)

**Scitrus 1.5.3** Настольная издательская система  
[bit.ly/29DkxWU](http://bit.ly/29DkxWU)

**Сайт: www.scitrus.net**

**SmartReView 1.27** Редактор изображений  
[bit.ly/29DkxWU](http://bit.ly/29DkxWU)

**Ubuntu 2017.06.13** Инструмент госадмина  
[bit.ly/29DkxWU](http://bit.ly/29DkxWU)

Описание на обложке »

## Информация о диске

### Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials [Главное]» на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

### Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

### Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

### Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя\_программы-1.0.1.i386.rpm** — вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя\_программы-1.0.1.i386.deb** — такой же пакет, но уже для Debian;
- » **имя\_программы-1.0.1.tar.gz** — обычно это исходный код;
- » **имя\_программы-1.0.1.tgz** — тот же файл, что и выше этажом по списку: «tgz» — это сокращение от «tar.gz»;
- » **имя\_программы-1.0.1.tar.bz2** — тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя\_программы-1.0.1.src.rpm** — также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя\_программы-1.0.1.i386.FC4.RPM** — двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя\_программы-1.0.1.ppc.Suse9.rpm** — двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя\_программы-devel-1.0.1.i386.rpm** — версия для разработчиков.

### Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: [disks@linuxformat.ru](mailto:disks@linuxformat.ru)

Внимательно прочтите это перед тем, как использовать LXF DVD!

НОВЫЙ РЕЛИЗ

# Debian 9.0

Исследуйте прочный как скала дистрибутив серверного класса для всех!

» Рабочий стол Gnome » Ядро 4.9 LTS » Новые средства анализа » MariaDB » PHP 7.0

64-битный

# Уоуагер Live 9.0

Дружелюбный дистрибутив Linux с предустановленным рабочим столом

НОВЫЙ РЕЛИЗ

» Ядро 4.9 » Рабочий стол Xfce » Интерфейс для Dock » Виджеты Solix

64-битный

**Linux**  
FORMAT

**LIVE-ДИСК: ГОТОВ К РАБОТЕ**  
ВСЕ НЕОБХОДИМОЕ ДЛЯ СТАРТА В LINUX

# Содержание



## ГЛАВНОЕ

Checkinstall  
Coreutils  
Hardinfo  
Kernel  
Memtest86+  
Plop  
SBM  
WVDial

## УЧЕБНИКИ

Загадка LXF: Вычисляем π  
Пишем Сору на Python

## ДОКУМЕНТАЦИЯ: 12 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

**Advanced Bash Scripting Guide** Подробное руководство по программированию на Bash  
**Bash Guide for Beginners** Руководство по Bash для начинающих

**Bourne Shell Scripting** Начальное руководство по программированию на Bash

**The Cathedral and the Bazaar** Классический текст

Эрика Реймонда [Eric S Raymond] «Сбор и базар»  
**The Debian Administrator's Handbook** Руководство администратора, написанное именитыми разработчиками Debian

**Dive Into Python** Учебник по программированию на Python

**Intro to Linux** Начальное руководство по Linux

**Linux Dictionary** Словарь Linux, объясняющий специфическую терминологию

**Linux Kernel in a Nutshell** Описание ядра Linux, созданное одним из его выдающихся разработчиков — Греггом Краа-Хартоманом [Greg Kroah-Hartman]

**Rule** Знаменитый справочник

**System Administrators Guide** Руководство по базовому администрированию Linux

**GNU Tools Summary** Руководство по работе в командной строке и обзор основных утилит GNU

## Страна 2

### БИЗНЕС-ДИСТРИБУТИВЫ

**ClearOS 7.0** (64-битный)

Дистрибутив для малого бизнеса, на базе CentOS, с серверными функциями и функциями сервера и шлюза. Хотя в нем присутствует веб-интерфейс, может полностью управляться из командной строки.

**Koozali 9.2** (64-битный)

Дистрибутив на базе CentOS, пригодный для размещения разных серверов сразу: файлового сервера, почтового сервера и сервера брандмауэра/шлюза. Бесплатно предлагает функции удаленного доступа и резервного копирования.

**NetServer 7.3** (64-битный)

Дистрибутив на базе CentOS, с модульным дизайном, благодаря чему легко превращается в почтовый сервер, фильтр, веб-сервер, брандмауэр. Предлагает веб-интерфейс администратора и установку «одним щелчком» ряда преднастроенных модулей.

**Zentyal 5.0** (64-битный)

Хорошо отлаженный дистрибутив на базе Ubuntu, с интуитивным веб-интерфейсом, вероятно облегчающим администрирование. Быстро разворачивается и настраивается как сервер шлюза, офисный сервер, унифицированный сервер коммуникаций.

### NOTICES

**Frogatto (Git)** Игра-аркада

**blivzibuster**

Все дистрибутивы представлены ISO-образом, который можно записать на отдельный носитель, и загрузить в live-режиме прямо с LXFDVD. У всех присутствует возможность установки на жесткий диск.

**Пожалуйста, перестаньте использовать формат АБННОГО ДИСКА ознакомьтесь с инструкцией, опубликованной в журнале на стр. 109!**

**КОММЕНТАРИЙ** Присылайте ваши пожелания и предложения по электронной почте: [info@linuxformat.ru](mailto:info@linuxformat.ru)

**ДЕФЕКТИВНЫЕ ДИСКИ** В маловероятном случае обнаружения дефектов на вашем диске, обращайтесь, пожалуйста, по адресу [diskdefect@linuxformat.ru](mailto:diskdefect@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 аппаритурный 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. Предоставляем наши программы или данные, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD. Предоставляем наши программы или данные. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, скачайте программу для загрузки с нашего сайта [www.linuxformat.ru](http://www.linuxformat.ru)

Тираж издательства ООО «Марком», 188652, Россия, Ленинградская область, Всеволожский р-н, дер. Юкки, Школьная ул., 7-а. Лицензия ИМПР ВАО N 77-03.

## Создание установочных дисков при помощи cdcrecord

Самый быстрый способ записать ISO-образ на чистую матрицу — это обратиться к программе *cdcrecord*. Для всех перечисленных ниже действий потребуются права суперпользователя-root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdcrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке — например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdcrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdcrecord*, сохраните некоторые настройки в файле `/etc/default/cdcrecord`. Добавьте по одной строке для каждого устройства записи (скорее всего, в вашей системе присутствует только одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке — метка; затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Для записи ISO-образа вам осталось набрать команду

```
cdcrecord -v /path/to/image.iso
```

Если вы не принадлежите к любителям командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее от имени root и выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

## Другая ОС?

Использовать Linux для записи компакт-диска не обязательно. Программы вроде *cdcrecord* просто переносят двоичные данные на чистую матрицу. Все необходимые файлы уже включены в ISO-образ, который распознается любой операционной системой, будь то Linux, Windows, Mac OS X или AmigaOS.

## Нет устройства для записи дисков?

Если у вас нет устройства, с помощью которого можно было бы записать образ на диск, можно найти какого-нибудь друга или организацию, у кого есть компьютер с дисководом, и прожечь диск у них. Опять-таки, вам подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт разработчика дистрибутива.

# LINUX FORMAT

# Подписывайтесь и читайте Linux Format на iPad или iPhone!

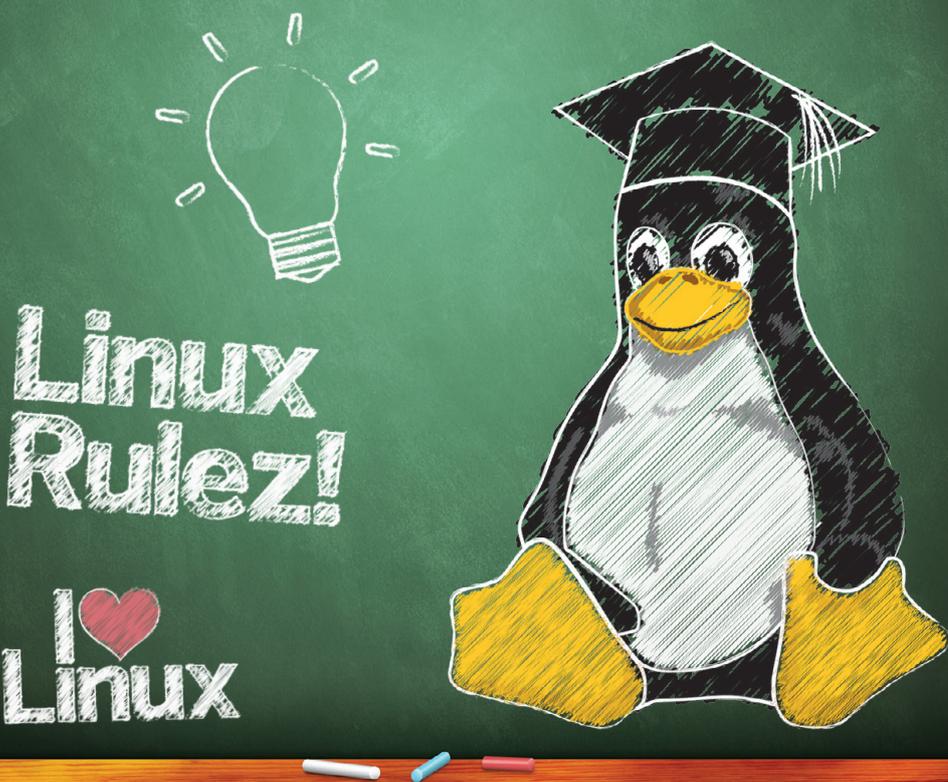
Доступно  
в AppStore



А если у вас Android, подпишитесь  
на Linux Format через Zinio!

 zinio™  
Доступно в Google Play





В октябрьском номере

Учебке поможет...

# Linux в школе!

Узнайте, как с помощью Linux ваши детки изучат в классе компьютеры на крутых проектах FOSS.

## Жизнь после Ubuntu Phone

Разберемся, что пошло не так с Ubuntu Phone и почему провалилась конвергенция от Canonical.

## Linux Multimedia Studio

Вместе сочиняем прекрасную музыку (нет, правда!) на профессиональном уровне при помощи комплекта LMMS для композиторов.

## Открытый телефон: мечты

Одни мечты умирают, но из унавоженной ими почвы произрастают другие, всё новые и новые.

## Файлы дышат в унисон

Выбираем лучшее средство синхронизации ваших файлов — на всех ваших устройствах.

Содержание будущих выпусков может меняться — вдруг накопленные знания взорвут нам мозг...

# LINUX FORMAT

Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № ФС77-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж печатной версии 1200 экз., распространение электронной версии 30000 экз.

### РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

#### Главный редактор

Кирилл Степанов [info@linuxformat.ru](mailto:info@linuxformat.ru)

#### Литературный и выпускающий редактор

Елена Толстякова

#### Переводчики

Елена Ессяк, Даниил Кривошеин, Светлана Кривошеина, Валентин Развозжаев, Валерий Смирнов, Елена Толстякова, Ирина Шулакова

#### Редактор диска

Александр Баракин

#### Верстка, допечатная подготовка

Сергей Рогожников

#### Технический директор

Андрей Смирнов

#### Директор по рекламе

Владимир Савельев [advert@linuxformat.ru](mailto:advert@linuxformat.ru)

#### Генеральный директор

Павел Фролов

#### Учредители

Частные лица

#### Издатель

ООО «Линукс Формат»

#### Отпечатано в типографии ООО «ЛД-ПРИНТ»

196644, Санкт-Петербург, Колпинский р-н, пос. Саперный, территория предприятия «Балтика», д. 6/н, лит. Ф  
Тел. (812) 462-8383, e-mail: [office@ldprint.ru](mailto:office@ldprint.ru)  
Заказ 16432

### РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор [Neil Mohr] [neil.mohr@futurenet.com](mailto:neil.mohr@futurenet.com)

Научный редактор Джонни Бидвелл [Jonni Bidwell]

[jonni.bidwell@futurenet.com](mailto:jonni.bidwell@futurenet.com)

Выпускающий редактор Клиффорд Хоуп [Cliff Hope]

[clifford.hope@futurenet.com](mailto:clifford.hope@futurenet.com)

Художественный редактор Эфраин Эрнандес-Мендоза

[Efrain Hernandez-Mendoza] [efrain.hernandez-mendoza@futurenet.com](mailto:efrain.hernandez-mendoza@futurenet.com)

### ПОДГОТОВКА МАТЕРИАЛОВ

Джонни Бидвелл [Jonni Bidwell], Нейл Ботвик [Neil Bothwick], Стюарт Бернс [Stuart Burns], Нейт Дрейк [Nate Drake], Лейф Джонсон [Leif Johnson], Джереми Лейрд [Jeremy Laird], Нейл Мор [Neil Mohr], Ник Пирс [Nick Peers], Лес Паундер [Les Pounder], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Александр Толстой [Alexander Tolstoy], Михалис Цукалос [Mihalis Tsoukalos], Рейчел Вонг [Rachel Wong], Евгений Балдин, Андрей Гондаренков, Дмитрий Пантеличев, Алексей Федорчук, Максим Черепанов

Иллюстрации Шейн Коллиндж [Shane Collinge]

Иллюстрация с обложки [www.magictorch.com](http://www.magictorch.com)

### КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Тел. +44 (0) 1604 251045, email: [linuxformat@myfavouritemagazines.co.uk](mailto:linuxformat@myfavouritemagazines.co.uk)

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

Тел. +7 (812) 309-0686, e-mail: [info@linuxformat.ru](mailto:info@linuxformat.ru)

По вопросам сотрудничества, партнерства, оптовых закупок:

[partner@linuxcenter.ru](mailto: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 ответственность за повреждение или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

"GNU/Linux" заменяется на "Linux" в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. [www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html)

За информацией о журнале, издаваемом Future plc group company, обращайтесь на сайт [www.futureplc.com](http://www.futureplc.com).



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

16+

# РОББО КЛУБ™

## Образовательные кружки для детей

Робототехника ● Программирование ● 3D-моделирование ● 3D-печать



### Обучение: от игры к профессии

для дошкольников

- ★ **Игровая робототехника**  
Возраст — от 5 лет.  
Требуется уверенный навык чтения

для школьников

- ★★ **Стартовый**  
Основы знаний
- ★★★ **Базовый**  
Специальность
- ★★★★ **Углубленный**  
Профессия, соревнования,  
собственные проекты

А ТАКЖЕ

- Праздничные мастер-классы
- Городской лагерь на время каникул
- **Курсы повышения квалификации для педагогов**



Инновационная  
программа обучения



Собственная  
методическая база



Российские технологии  
и оборудование

## ВООБРАЖАЙ! ПРОГРАММИРУЙ! СОЗДАВАЙ!



[www.robboclub.ru](http://www.robboclub.ru)

8 800 777 2985

[vk.com/robboclub](https://vk.com/robboclub) ● [info@robboclub.ru](mailto:info@robboclub.ru)

**HETZNER**  
ONLINE

**БЫСТРЫЙ КАК МОЛНИЯ**

**ОСТРЫЙ КАК ЛЕЗВИЕ**

**НОВИНКА**



## Выделенный сервер EX51-SSD-GPU

Intel® Core™ i7-6700 Quad-Core Skylake Processor  
64 ГБ DDR4 RAM  
2 x 500 ГБ SATA 6 Gb/s SSD  
Видеокарта GeForce® GTX 10800  
100 ГБ место для резервных копий  
50 ТБ трафик\*  
Без минимального контракта  
Установка 7200 рублей



**7200** рублей в месяц

## Идеальное решение для ресурсоёмких вычислений.

Новый выделенный сервер EX51-SSD-GPU содержит невероятно мощную графическую карту GeForce® GTX 1080 для ускорения графических приложений и быстрого 3D-рендеринга.

[www.ru.hetzner.com](http://www.ru.hetzner.com)

\* Нет платы за превышение. При превышении 50 ТБ/месяц скорость соединения ограничивается (подсчёт ведётся по исходящему трафику, входящий и внутренний трафик не учитывается). Опционально можно снять ограничение, подтвердив оплату 85 руб. за каждый дополнительный ТБ.

Все цены вкл. НДС 18%. Цены могут измениться без уведомления. Все права защищены соответствующими производителями. Intel, логотип Intel, Core и Core Inside являются товарными знаками корпорации Intel в США и других странах.