

DVD ДАРОМ!



FEDORA 23 & KALI 2.0

# LINUX FORMAT

Главное в мире Linux

Апрель 2016 № 4 (208)

66 страниц  
учебников  
и статей!

- » Мультизагрузка всего!
- » Ubuntu для новичков
- » Облачное хранилище Hadoop



Академия кодинга:  
Принимаемся за Swift!

# ТАЙНЫ ХАКЕРОВ

Технология бронезащиты от атак  
и безопасности онлайн



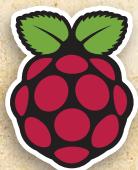
- » Напрочь отгоним хакеров
- » Защитим свои серверы
- » Обнаружим эксплойты
- » Тестируем на Android, Pi и не только!



## Бойтесь прошивок

« АНБ может заставить Intel внедрить уязвимость в их процессоры »

Мэттью Гэрретт — об ужасах прошивок с. 38



## Плюс: Для Pi!

- 8-страничный гид по Raspberry Pi
- » Кодируем на Ruby и на Pi
- » Сервер документов OwnCloud 8.0
- » Pi Zero·Explorer·PHAT



Сравнение

## Кодирование видео

» Тестируем самые шустрые транскодеры для высокого разрешения



База данных NoSQL

## Рулим MongoDB

» Освоим любимую Интернетом базу данных



ПОДПИСНЫЕ ИНДЕКСЫ В КАТАЛОГАХ

Агентство «Роспечать» — 36343,

«Почта России» — 11932, «Пресса России» — 90959

Linux center  
[www.linuxcenter.ru](http://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

Вопрос нашим экспертам был таким: в данном номере мы изучаем хакерские секреты — каков ваш самый большой или любимый секрет Linux, который, по-вашему, должен знать каждый?



### Джонни Бидвелл

В меню настройки ядра, если вы ищете символ и нажимаете номер, соответствующий ему в результатах, вы попадаете прямо на него, а не ищете путь к нему сами. Ну да, при поиске работы почти то же самое, но до недавнего времени эта жемчужина от меня ускользала.



### Ник Пирс

Вот вам хак иного рода — я его открыл, собирая из Raspberry Pi Zero приемник AirPlay для журнала *Mac Format*. Я скомбинировал его с ЦАП за £15 от <http://igaudio.co.uk> и блестящим дистрибутивом Pi MusicBox — и получил отпадно-потрясающий звук!



### Лес Паундер

Я люблю способ поиска в предыстории *Bash*. Я часто пользуюсь сочетанием клавиш Ctrl+г для интерактивного поиска команды в моей истории, чтобы не шарить там с помощью *grep*. И можно повторить команду, найдя ее запись в истории и введя ее номер с префиксом '!'.



### Маянк Шарма

Вообще-то это не секрет, но магия клавиши SysRq, вероятно, утрачена для многих новых пользователей Linux, тем более что сама клавиша из многих ноутбуков исчезла. По клавише SysRq/PrtSc можно передать команду прямо в ядро Linux, что обычно применяется для аккуратного перезапуска зависшей машины.



### Валентин Синицын

Enter, тильда, точка прерывает зависшую намертво сессию SSH. Этот простой трюк сэкономил мне уйму времени. Узнайте больше в Enter, тильда, знак вопроса.



## Угрозы и риски

» Хакерские атаки, кража данных, DDoS — панические сообщения о таких происшествиях появляются в новостях чуть ли не каждую неделю. Кажется, что мир стоит на грани катастрофы, и задача обеспечения информационной безопасности практически не решаема. Так ли это?

С одной стороны, мы действительно не можем гарантировать отсутствие уязвимостей в программном обеспечении и аппаратуре. Можно говорить только о вероятностях. Админы тоже не идеальны. Именно их ошибки чаще всего и приводят к печальным последствиям. А от «отказа в обслуживании» вообще никто не застрахован. Любую систему можно перегрузить запросами, любой ее компонент может банально сломаться.

С другой стороны, теория учит нас, что риск ИБ возникает только тогда, когда источник угрозы может с той или иной вероятностью использовать уязвимость. Эту вероятность можно и должно уменьшать. Вопрос только в «ценах вопроса». Вот тут-то и приходится вычислять риск — сумму этих двух вероятностей, умноженную на величину ущерба. Опыт показывает, что последнюю многие склонны преувеличивать. Далее все просто — затраты на защиту должны быть адекватны риску. Не все угрозы актуальны, не все риски неприемлемы. Главное — без паники!

### 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/linuxformat](http://vk.com/linuxformat)

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

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

# ПОДПИШИСЬ НА LINUX FORMAT!

Оформи подписку на печатную версию журнала Linux Format на 2016 год и получи в подарок диск с архивом номеров 2005—2014, а также подписку на электронную версию в формате PDF



Приложение  
для чтения  
**Linux Format**  
на iPhone и iPad  
доступно  
в AppStore

## SHOP.LINUXFORMAT.RU

### Варианты доставки

- » Почтой по России простой или заказной бандеролью
- » Курьерской службой СПСР или IML
- » В виде PDF-файлов для подписчиков электронной версии

### Редакция

- » Санкт-Петербург: пр. Медиков, 5, корп. 7,  
(812) 309-0686
- » Москва: (499) 271-4954



# Доверяйте свой контент только профессионалам

Сеть доставки контента CDNvideo с 2010 года обеспечивает качественным, разнообразным и бесперебойным сервисом клиентов в России, СНГ, Европе и Юго-Восточной Азии.

## Бесперебойные онлайн-трансляции



Высокая скорость и качество видео  
Вещание на неограниченную аудиторию  
Корректное отображение на любых устройствах  
Множество дополнительных опций

## Мультимедиа по запросу



Стабильно высокая скорость загрузки  
Без буферизаций и зависаний  
Не требует капитальных затрат  
Снижение нагрузки на каналы связи

## Ускорение интернет-сайтов



Улучшает позиции в поисковой выдаче  
Положительно влияет на конверсию  
Повышает лояльность пользователей  
Улучшает пользовательский опыт в регионах

Подробности на сайте <http://cdnvideo.ru> и по телефону +7 495 212-02-46

CDNvideo – ведущий провайдер услуг сети доставки контента (CDN) в России и СНГ. Ёмкость сети составляет более 500 Гбит/с, в том числе в Москве – более 300 Гбит/с. Доступность сети в 2013-2015 гг. составила более 99,99%.

**Надёжность. Стабильность. Качество.**

+7 495 212 02 46  
[info@cdnvideo.ru](mailto:info@cdnvideo.ru)

# Содержание

«Это не я не умею хранить секреты, а те, кому я их доверяю.» Авраам Линкольн

## Обзоры

**Deepin 15 .....** 14

Дистрибутив на базе Ubuntu перекочевал на Debian и на этом как минимум не програл. Удобен в использовании и не запутает новичков.



➤ Отход от эстетики Ubuntu явно не повредил рабочему столу Deepin.

**Solus 1.0 .....** 15

Новичкам в этом месяце повезло — вот им еще один впечатляющий дистрибутив. Даже мы его зауважали.

**Da Vinci 1.0 Jr. .....** 16

Невероятно дешевый 3D-принтер вполне может поспорить по качеству даже с главными игроками на этом поле.

**Acer R11 .....** 17

Легким движением руки вы сможете превратить этот хромбук в планшет, благодаря шарниру на 360°.



➤ Крутится во все стороны! И не впервые, а все-таки круто.

**Lenovo Chromebox Tiny .....** 18

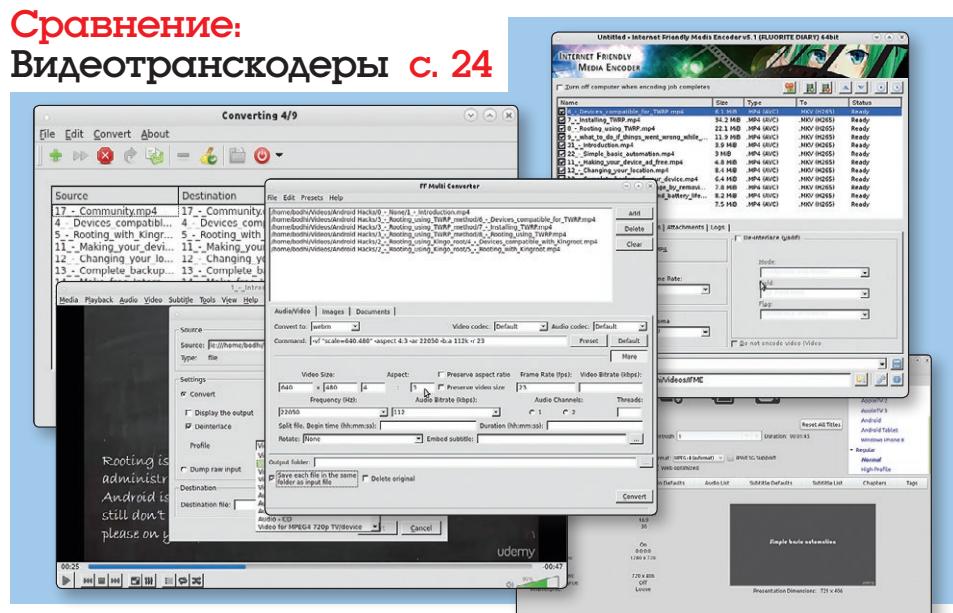
Все помешались на хромбуках, и Lenovo решила тоже установить ChromeOS. И получился отличный миниатюрный ПК, впору хоть и для офиса.

**Football Manager 2016 .....** 19

Время заняться футбольным менеджментом! Не так чтобы глобальным, но хотя бы в микроварианте...



## Сравнение: Видеотранскодеры с. 24



## Интервью

«Процесс, запущенный обычным пользователем, вдруг получит права root.»

Мэттью Гэрретт — об ужасах эксплойтов в прошивках с. 38

# На вашем бесплатном DVD



## Fedora Security 32-битный, Kali 2.0 Light 64-битный, Manjaro 15 64-битный

» Каждый месяц — только лучшие дистрибутивы

ПЛЮС: GParted и многое другое! с. 106

Побалуйте себя и любимых  
подпиской на LXF!

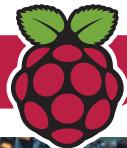


Доступно  
в AppStore!



[www.linuxformat.ru/subscribe](http://www.linuxformat.ru/subscribe)

## Пользователям Raspberry Pi



### Новости Pi ..... 88

Помогите детям в Mali и оцените ситуацию с Pi во Франции, а также мощь кластера Pi Zero.

### Explorer Pi HAT ..... 89

Мы поревелись с замечательной новейшей приставкой к Pi Zero — отличная штука!

### Ruby и Sonic Pi ..... 90

В этом Sonic Pi скрывается Ruby; ну как не поэксплуатировать подобный факт?

### OwnCloud 8.0 ..... 92

Изучаем новые свойства облачного сервера ownCloud и настраиваем на Pi свой собственный.



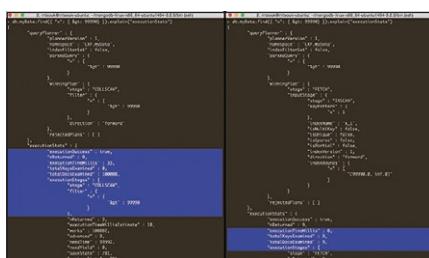
## Академия кодинга

### Букварь по Swift ..... 80

Apple всех удивила: создала свой язык Swift и выдала его в открытый доступ. Поговаривают, что языку PHP после этого остается только нервно курить в сторонке, горюя о невостребованности.

### MongoDB для админов ..... 84

Ваш личный сервер NoSQL с точки зрения администрирования: приобщаемся к основам. Запросы, индексы, репликация... и загадочное шардирование.



## Постоянные рубрики

### Новости ..... 6

Samsung братает Android и Windows, Linux шагает на ноутбуки, атаки посредника блокированы, WD купила SanDisk, Ubuntu скрестили с BSD и заснули в Windows, Intel выдала новые SSD-накопители, а суперкомпьютеры специализируются.

### Мобильные новости ... 20

Samsung возражает против фрагментации и выпускает SSD-накопители для смартфонов, Android пришел на ПК, а BlackBerry отменяет абонентскую плату.

### Сравнение ..... 24

Вам надо перекодировать видеофайлы? Мы припасли для вас кое-что: FF Multi Converter, Handbrake, iFME, QwinFF, VLC Media Player.

### Интервью LXF ..... 38

Мэтью Гэрретт предупредил: пока не открыты драйверы и прошивки устройств, безопасности не будет.

### Рубрика сисадмина .... 46

М-р Джолион Браун демонстрирует, что настройка контейнерной инфраструктуры — это вовсе не больно, и сообщает, что сертификация Let's Encrypt вошла в полную силу. А также про ошибку OpenSSH.

### Ответы ..... 96

#### ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!

Нейл Ботвик — про установку ремикса Ubuntu, начало работы с Tails Linux, обновление с Ubuntu LTS, перенос Grub из MBR в загрузочный раздел, подключение Android к Linux, выбор между метками разделов и UUID.

### HotPicks ..... 100

Отведайте горяченько! Лучшие в мире новинки свободного ПО: Binaria, Boxes, Let's Encrypt, Mixxx, MP3fs, Mumble, PeaZip, Qemu, Sprout.io, SuperTux, Zeal.

### Диск Linux Format ..... 106

Содержимое двустороннего DVD этого месяца.

### Пропустили номер? ... 108

Ну вот, вы не узнали, как избавиться от Windows 10... но это пока еще поправимо.

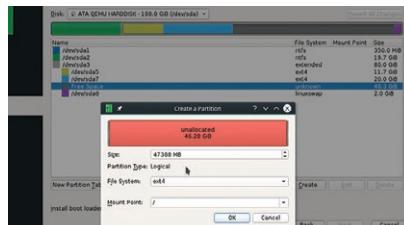
### Через месяц ..... 112

Linux повсюду — от дронов до телефонов и светофоров. Посмотрим, что за интересные устройства на нем работают.

## Ищите в номере

### Загружаем всё! ..... 42

Двойная загрузка? Фи, отстой из 2015 года. Модный тренд — мультизагрузка! Ставим столько ОС, сколько душе угодно.



» Запихиваем дистрибутивы пачками.

## Учебники

### Ключевые утилиты

#### Сортируем ..... 52

Это нам запроси — хоть по алфавиту, хоть по возрастанию.

#### Рабочий стол Unity

#### Управляем иконками ..... 56

Приведем в порядок иконки на рабочем столе, чтобы они радовали глаз и концентрировали вас на главном.

#### Поиск неисправностей

#### Запуск Ubuntu ..... 58

Разберемся с неприятностями при загрузке системы и истребляем их.

#### Gentoo

#### Portage — это просто ..... 62

Отметим обвинения в том, что мы пишем руководства только для Ubuntu.

#### Отладка среды

#### Procev ..... 64

Изучим удобнейший инструмент отладки, который помогает привести в чувство среду разработчика.

#### Облачное хранение

#### Hadoop ..... 68

Перекидываем свои вычисления по Большшим Данным в облако и обратно.

#### MySQL Fabric

#### Работа с шардами ..... 72

Принимаемся за ферму серверов не по-детски — создаем схему шардирования.

#### IP-телефония

#### Диалплан и диалог ..... 76

Изящно завершаем серию статей набором номеров и настройкой автоответчика.

# Новости

**В ЭТОМ НОМЕРЕ:** Связь смартфонов » Linux на ноутбуках » Прослушка затрудняется » Удачное поглощение » Ubuntu с ядром BSD » Ubuntu... в Windows! » Супер-диски » HP специализируется

## НАВОДИМ МОСТЫ

# Планшет с функцией Flow

Samsung пытается преодолеть стену между Android и Windows.

**F**low — новая функция Galaxy Tab-Pro S — позволит входить в систему на этом планшете с помощью смартфона. Samsung, вслед за Apple, хочет устранить разрыв между своими Android-смартфонами и устройствами Windows. Благодаря Flow планшет с Windows 10 сможет эффективно работать с Android-смартфонами Galaxy. Используя сканер отпечатков пальцев своего смартфона, пользователь сможет войти в систему TabPro S, где такого сканера нет. Войти в TabPro S можно будет и с помощью комбинации знаков для аутентификации на смартфоне. Приложение Flow, установленное на смартфоне, по протоколу NFC передает на TabPro S информацию об аутентификации; Flow позволяет уведомлениям о телефонных звонках и текстовым сообщениям, полученным смартфоном Galaxy, отображаться

также и на TabPro S — как у «умных» часов Samsung Gear; наконец, Flow связывает TabPro S с мобильной точкой доступа Wi-Fi на смартфоне Galaxy.

Дисплей TabPro S может отображаться на экране Samsung TV — смартфоны Samsung давно научились управлять телевизорами, а также «умными» холодильниками и стиральными машинами компаний. Но попытка совместной работы Windows и Android создает проблемы, из-за различий в кодах и протоколах. «С Flow связана масса возможностей, но Samsung не определил, какие добавочные функции в него встроят», — сказал пресс-секретарь компании на пресс-конференции в Нью-Йорке.

TabPro S — первый планшет Galaxy на Windows. В США поставки начались 17 марта, цена модели начального уровня, \$899, такая же, как у базовой Microsoft



» Рубрику готовил  
**АНДРЕЙ  
ГОНДАРЕНКОВ**

Surface Pro 4 с аналогичной конфигурацией, с чипом Intel Core M3 и накопителем 128 ГБ. Толщина планшета 6,35 мм, вес 680 г. Изюминка планшета — дисплей Super AMOLED с разрешением 2160×1440 пикс. У OLED-экранов более «сочные» цвета и лучше энергоэффективность, так как у них отсутствует подсветка.



» Galaxy Tab Pro S:  
ответ Samsung  
на тренд two-in-one,  
начатый Microsoft  
Surface Pro.

## НОУТБУК ДЛЯ LINUX

# Стратегия Dell по Linux-ПК

Ноутбуки Project Sputnik стимулируют к переходу на Linux с Mac и Windows.

**И**зящный дизайн нового ноутбука от Project Sputnik, Dell XPS 13 Developer Edition, заставляет пересмотреть мнение о невзрачности ноутбуков с Linux. Задействованные в XPS 13 DE технологии — экраны 4K, процессоры Intel Skylake, высокоскоростной интерфейс Thunderbolt 3 — уже привычны в Mac OS и Windows, но на ноутбуках с Linux пока еще в новинку. XPS 13 с Ubuntu 14.04 LTS — кузен вышедшего ранее в этом году такого же ноутбука с Windows 10. Выпустить обе модели одновременно не удалось из-за

неготовности драйверов для Linux; отстала также поддержка чипов Skylake в Ubuntu, заявленная только 18 февраля. При этом XPS 13 DE не ограничивается лишь Ubuntu, Dell может сертифицировать его работу и с другими дистрибутивами Linux.

Сейчас Dell работает над поддержкой док-станции для Linux-версии XPS 13 DE, что обеспечит дополнительные порты расширения (у Windows-версии поддержка док-станции уже есть). А вот такую отличную функцию, как Windows Hello в Windows 10, для входа в систему посредством

биометрической аутентификации, Dell сможет реализовать в Linux-версии только когда ее встроят в ОС разработчики.

Project Sputnik, задуманный 4 года назад ведущим инженером СТО Dell и энтузиастом Linux Бартоном Джорджем [Barton George], призван ускорить адаптацию новых аппаратных технологий в ноутбуки с Linux. Dell — крупнейший производитель ПК с такой масштабной поддержкой Linux. Ноутбуки с Linux предлагают и другие производители, скажем, System76, но самы широкий охват обеспечивает именно Dell.

ПОД ПРОТОКОЛ

# Механизм SMTP Strict Transport Security

Трафик электронной почты получил шифрование, неуязвимое для атак «посредника».

**C**отрудники ряда крупнейших мировых провайдеров электронной почты объединились в попытке улучшить безопасность почтового трафика, проходящего через Интернет. Разработанный инженерами Google, Microsoft, Yahoo, Comcast, LinkedIn и 1&1 Mail & Media Development & Technology механизм SMTP Strict Transport Security, позволяющий поставщикам услуг электронной почты определять политику и правила для создания зашифрованных сообщений email, сформулирован в проекте, опубликованном в средине марта для рассмотрения в качестве стандарта Internet Engineering Task Force (IETF).

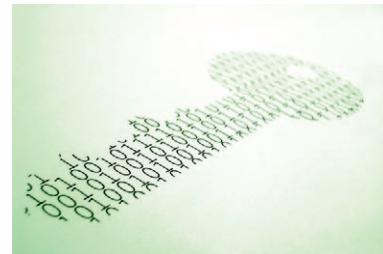
Ныне действующий протокол SMTP восходит к 1982 г. При его создании никакие опции шифрования не предусматривались, поэтому в 2002 г., как способ включить TLS (Transport Layer Security) для SMTP-соединений, в протокол добавили расширение

STARTTLS. В течение следующего десятилетия это расширение широкого распространения не получило, и трафик email оставался в основном незашифрованным.

Ситуация начала меняться после 2013 г., когда через бывшего контрактора АНБ США Эдварда Сноудена [Edward Snowden] просочились секретные документы, свидетельствовавшие о широкомасштабном наблюдении за интернет-коммуникациями спецслужбами США, Великобритании и ряда других стран. Тестирование, проведенное в мае 2014 г. Facebook, где ежедневно отправляются миллиарды email, показало, что только 58% этих писем прошли через соединение, шифруемое STARTTLS. Зато в августе эта величина составила уже 95%.

Однако, в отличие от HTTPS (HTTP Secure), STARTTLS допускает так называемое «оппортунистическое» шифрование (OE): достоверность цифровых сертификатов серверов email не проверяется, т. е.

По данным Google, шифруются 83% исходящих сообщений от пользователей Gmail к другим провайдерам email, но только 69% входящих сообщений извне поступают по зашифрованному каналу.



такие соединения уязвимы к атакам man-in-the-middle [человек-посредник]. Кроме того, соединения STARTTLS уязвимы к атакам «encryption downgrade [деградация шифрования]».

SMTP STS решает обе проблемы. Политики SMTP STS определяются с помощью специальных DNS записей, добавляемых к доменному имени сервера email. Для клиентов протокол предусматривает механизмы автоматической проверки этих политик и предоставление информации о сбоях.

СДЕЛКИ

## WD приобрела SanDisk

Ожидаем инновационных решений по хранению данных.

**W**estern Digital Corp. (WD) объявила, что ее акционеры одобрили приобретение SanDisk Corp за \$19 млрд: более 90% акционеров компании голосовали за размещение простых акций. Слияние одобрили 98% акционеров SanDisk, и этот процесс, как ожидается, будет завершен во втором квартале текущего года.

SanDisk оценивается в \$17 млрд, и является основным поставщиком продуктов NAND flash memory, таких как карты SD и твердотельные накопители (SSD). В октябре минувшего года Western Digital первой объявила о своем намерении приобрести компанию SanDisk, предупредив, что выплаты будут осуществляться наличными и акциями.

Western Digital тоже поставляет продукты NAND flash, однако в первую очередь

это производитель жестких дисков (HDD) и ПО для управления. По мнению вице-президента IDC по исследованиям Джека Януковича [Jeff Janukowicz], поглощение SanDisk позволяет Western Digital мгновенно занять прочное положение на мировом рынке памяти NAND flash.

Компании ИТ-индустрии изыскивают способы охватить сферы Интернета вещей, носимых устройств и «облаков», и в результате в последнее время наблюдается волна слияний, поглощений и инвестиционной активности на рынке систем хранения данных. Соглашение WD и SanDisk следует вплотную за самой крупной из когда-либо заключенных сделок на рынке хранения данных — покупкой Dell компанией EMC за \$67 млрд. Кроме того, компания-производитель полупроводниковых устройств хранения PMC-Sierra также получила



Акционеры WD и SanDisk одобрили слияние двух компаний: сумма сделки составила \$19 млрд в ценных бумагах и наличными.

множество предложений о поглощении, а компания Unisplendour (Синьхуа, Китай) согласна приобрести 15% Western Digital за \$3,78 млрд.

## ГИБРИДНЫЕ ОС

# Симбиоз Ubuntu Linux и ядра BSD

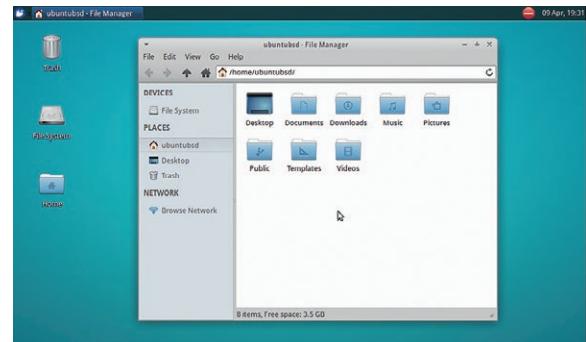
«В одну телегу впрячь не можно коня и трепетную лань...»?

**U**buntuBSD — новая операционная система с открытым исходным кодом, объединяющая Ubuntu Linux с ядром FreeBSD и интерфейсом *Xfce*.

BSD — открытая Unix-подобная ОС, долгое время остающаяся в тени Linux — вошла в мир Ubuntu благодаря новому open-source проекту UbuntuBSD. Напомним, что создание BSD началось в конце 1970-х гг. (первоначально в качестве расширенной версии ОС Unix от AT&T, а затем и полной ее замены) в Университете Калифорнии (Беркли); это была одна из первых свободно распространяемых ОС. В 1990-х, по разным сложным причинам — в основном, но не исключительно, связанных с юридическими проблемами — BSD оставалась позади других свободных ОС, построенных на ПО GNU и новом тогда ядре Linux. Несколько BSD-based ОС сохранились и в настоящее время: FreeBSD, NetBSD и OpenBSD, но они никогда не имели такого влияния на рынке серверных и настольных систем, как системы GNU/Linux.

Теперь же группа программистов намерена объединить один из самых популярных дистрибутивов GNU/Linux, Ubuntu от компании Canonical, с BSD в новой ОС под достаточно внятным названием UbuntuBSD. Идея проекта — «реализовать удобство и дружелюбие Ubuntu с незыблевой стабильностью и производительностью ядра FreeBSD». UbuntuBSD вряд ли станет для большинства такой же привлекательной, как Ubuntu — хотя бы потому, что среди по умолчанию служит *Xfce*, а не Unity (т.е. UbuntuBSD более походит на объединенный с ядром BSD движок Ubuntu сальтернативным интерфейсом). Но тем, кому нужна настольная среда, менее требовательная к ресурсам, чем Unity и Gnome, но симпатично выглядящая и обеспечивающая широкую функциональность, новая ОС придется по душе.

В середине марта UbuntuBSD насчитывала уже почти 5 тыс. загрузок, хотя оставалась в виде бета-версии, еще не готовой к продуктивному использованию. Проект



➤ **UbuntuBSD:**  
рабочий стол —  
*Xfce*, файловая  
система — *ZFS*.

интересен уже тем, что обеспечивает новые возможности для интегрирования обширной экосистемы ПО, сложившейся вокруг Ubuntu, с ядром, предлагающим ряд отсутствующих в Linux возможностей. Это не говоря о более либеральных условиях лицензирования, т. к. ядро FreeBSD не использует лицензии GNU GPL. Все это может породить в дальнейшем не только новый тип настольных ОС, но также новые серверные или облачные решения.

## НЕ МОЖЕТ БЫТЬ!

# WSL: Ubuntu в Windows

Обновление Windows 10 позволит пользователям запускать внутри себя Ubuntu.

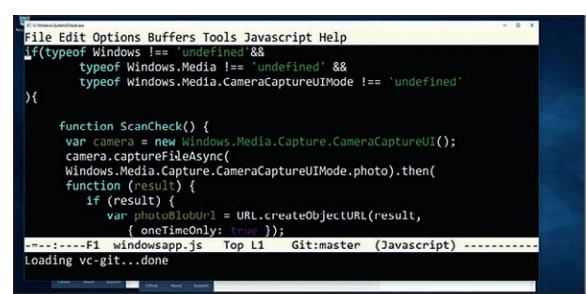
**В** очередном крупном обновлении Windows 10, Redstone, Microsoft предоставит пользователям возможность запускать в родном формате командную оболочку *Bash* и двоичные файлы Ubuntu. Причем не в контейнере или виртуальной машине, но с помощью нативных библиотек и программ Windows новой инфраструктуры Windows Subsystem for Linux (WSL).

WSL была помещена в код Windows 10 (build 14251) еще в конце января. Несколько дней спустя были добавлены две новые подсистемы, *Ixcore.sys* и *Ixss.sys*, ориентированные на Windows-программистов, разрабатывающих приложения для Linux. Но основное предназначение WSL — поддержка образа Ubuntu в режиме пользователя.

Представитель от Canonical Дастин Керклэнд [Dustin Kirkland] подчеркивает, что речь идет не о перекомпилированных открытых утилатах Cygwin, а о запуске

двоичных ELF-файлов Ubuntu прямо в Windows, bit-for-bit и checksum-for-checksum. Основной трудность был перевод в режиме реального времени системных вызовов Linux в системные вызовы Windows. Умники Linux могут представлять это как «*WINE* наоборот» (нет, WSL еще не open source). Утилита *sysbench* показывает практически эквивалентную производительность CPU, памяти и I/O. Тем не менее, Canonical характеризует нынешнее состояние WSL как бета-релиз: «недостатки касаются в основном *tty* и *vt100*. Не полностью функциональны пока также *screen* и *tmux*».

Керклэнд напомнил, что Ubuntu — наиболее популярный дистрибутив Linux в Azure и других «облачных» сервисах. Нативная оболочка Ubuntu, встроенная в рабочий стол Windows, позволяет сильно упростить запись кода с помощью Visual Studio, *vim* или *emacs*, последующее перемещение его в «облачные» реализации



➤ **Командная  
оболочка *Bash*,  
работающая  
в Windows 10.**

через *git*, *scp* или *rsync* и обратно. Многие из этих реализаций будут именно Azure Ubuntu. Скоро появится первый образ Ubuntu для Windows 10 — Ubuntu 14.04 LTS. Сменяющий его Ubuntu 16.04 LTS попадет в Windows Store вскоре после релиза 21 апреля (образ зависит от кода Redstone, и ему нужна сборка Windows Build 14251 или новее). Microsoft уже анонсировала релиз Windows 10 Redstone 1 ("Anniversary Update" или Windows 10 SP1) на нынешнее лето.

# Добро пожаловать в робототехнику!

# ScratchDuino

Электронный комплекс на основе свободного аппаратного обеспечения для школ и вузов

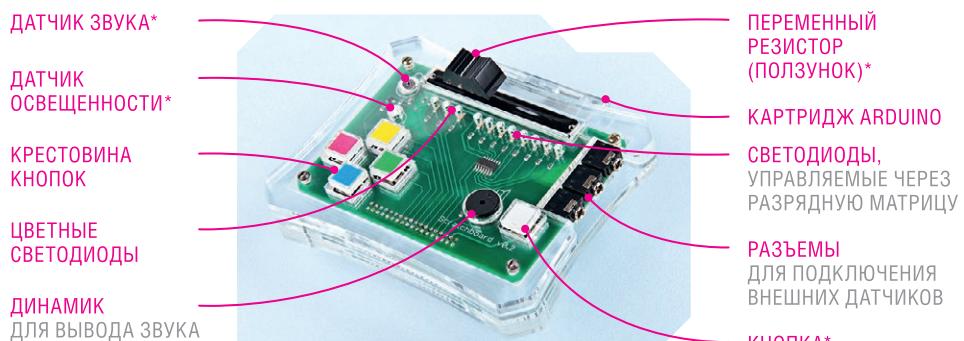
## ScratchDuino.Лаборатория

Плата расширения с датчиками ввода-вывода — как установленными на плате, так и подключаемыми. Предназначена для программирования взаимодействия компьютера с внешними устройствами.

Полная интеграция («из коробки») со средой программирования Scratch, предназначеннной для детей.

ЦЕНА 9500 руб.

ГАРАНТИЯ 3 ГОДА,  
ВКЛЮЧАЯ ТЕХНИЧЕСКУЮ  
И МЕТОДИЧЕСКУЮ  
ПОДДЕРЖКУ



\* РАБОТАЮТ В РЕЖИМЕ ЭМУЛЯЦИИ PICOBORD, НЕ ТРЕБУЯ ДОПОЛНИТЕЛЬНОГО ПРОГРАММИРОВАНИЯ.

## ScratchDuino.Робоплатформа

Внешний робот-исполнитель, управляемый из среды программирования Scratch. Не требует навыков программирования на языках высокого уровня и может применяться для обучения, начиная с младших классов.

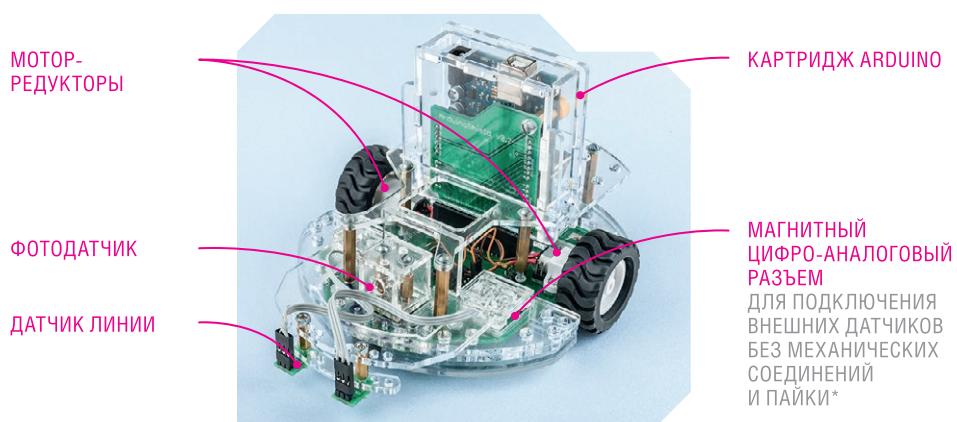
ScratchDuino.Робоплатформа

обеспечивает возможность:

- подключения внешних датчиков без механических соединений и пайки;
- расширения через последовательную шину RoboBus;
- установки деталей Lego Technics.

ЦЕНА 19 500 руб.

ГАРАНТИЯ 3 ГОДА,  
ВКЛЮЧАЯ ТЕХНИЧЕСКУЮ  
И МЕТОДИЧЕСКУЮ  
ПОДДЕРЖКУ



\* ВСЕГО НА РОБОПЛАТФОРМЕ 5 РАЗЪЕМОВ. ВОЗМОЖНО ПОДКЛЮЧЕНИЕ ДАТЧИКОВ КАСАНИЙ, ИК-ДАТЧИКОВ РАССТОЯНИЙ/ПРЕПЯТСТВИЙ И ДР.

### Назначение ScratchDuino

- Изучение взаимодействия компьютера с внешней средой
- Изучение изменения параметров внешней среды
- Изучение процессов передачи информации и принципов ее построения
- Изучение внешних устройств управления
- Моделирование устройств

### Среда разработки

Lazarus (язык Pascal) ■ Scratch ■ Arduino IDE

ScratchDuino адаптирован  
для учебных заведений.  
Поставляется с комплектами  
учебно-методических  
материалов.

Продукт разработан при финансовой поддержке Фонда содействия  
развитию малых форм предприятий в научно-технической сфере

[WWW.SCRATCHDUIANO.RU](http://WWW.SCRATCHDUIANO.RU)

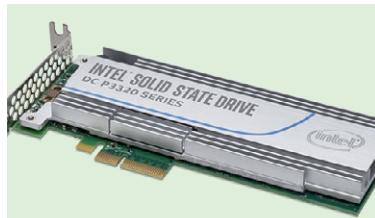
(812) 309-0686 (Санкт-Петербург) ■ (499) 271-4954 (Москва)

## ДЛЯ ХРАНЕНИЯ ДАННЫХ

# SSD-диски Intel с чипами 3D NAND

Быстрые, живучие и надежные накопители порадуют предприятия и data-центры.

Intel 31 марта объявила о готовности первого сверхбыстрого SSD-накопителя большой емкости (до 2 ТБ) с чипом 3D NAND. По емкости Intel пока отстает от Samsung, который в марте уже представил SSD-накопитель PM1633a на 15,36 ТБ, тоже на чипах 3D NAND. Новая линейка накопителей Intel включает модели SSD DC P3320, SSD DC 3520, SSD DC D3700 и D3600, и ориентирована на работу в data-центрах, рабочих станциях и массивах хранения данных. 3D NAND flash отличается большей плотностью записи: ячейки памяти размещены слоями друг над другом — в отличие от традиционных SSD, в которых flash-чипы размещены рядом. Чипы памяти также плотнее скомпонованы, отчего SSD-накопители быстрее. И подключаются эти SSD к слотам PCI-Express 3.0, которые бы-



стнее обычных SATA (протокол хранения PCIe формальное именуется NVMe — Non-Volatile Memory Express). Скорость чтения при произвольном доступе [random read speed] модели DC P3320 — 365K IOps, а random write speed 22K IOps. Скорость последовательного чтения [sequential read speed] — до 1600 МБ/с, скорость записи — 1400 МБ/с. Цена накопителя пока не назначена, а поставки начнутся во II квартале.

➤ Intel SSD DC P3320 — первый SSD компании с чипами 3D NAND.

## ОТ ОБЩЕГО К ЧАСТНОМУ

# Специализированные системы HPE

Эра суперкомпьютеров «на все случаи жизни» прошла.

В апреле Hewlett-Packard Enterprise (HPE) анонсировала серию новых систем для конкретных процессов, таких как «глубинное обучение [deep learning]». Решение HPE о создании высокоеффективных специализированных компьютерных систем вызвано в т.ч. ростом потребностей систем обработки данных. У нового сервера Apollo 6500 от HPE, который представят в III квартале, два сокета CPU и поддержка до 8 высокопроизводительных карт с GPU Nvidia. В «глубинном обучении» почти всю работу выполняет GPU, и в новой системе HPE с целью повышения эффективности их соотношение с CPU составляет 4:1 (стандарт — 1:1). HPE также создает HPC — спецсистемы для коммерческих приложений. Исторически высокоеэффективные системы ориентировались не на обработку данных,

а на вычисления: CPU был самой быстрой частью системы, общая эффективность которой зависела от того, насколько остальные составляющие могли не отставать от процессора. При обработке математических и коммерческих процессов применяется однопоточный алгоритмический код, и наличие нескольких процессорных ядер, большинство которых не будет использоваться, излишне. Trading-процесс эффективнее выполнять на односокетном сервере, при 8 ядрах CPU вместо 20. Это уменьшает накладные расходы и позволяет повысить тактовую частоту.

HPE — крупнейший вендор HPC, и его действия неизбежно будут влиять на рынок. В последнем Top-500 системам HPE принадлежит 31%, наибольшая часть глобального сектора. Вторую позицию занимает Lenovo с 13,8%. **LXF**

## Новости короткой строкой

➤ В Единый реестр российских программ для ЭВМ и баз данных включены дистрибутив Alt Linux Школьный и Astra Linux Special Edition. Источники: [www.altlinux.ru](http://www.altlinux.ru) и [www.astralinux.com](http://www.astralinux.com)

➤ В обновление Android Auto 1.6 добавлена поддержка 18 новых стран, включая Россию. Источник: [twitter.com](http://twitter.com)

➤ В рамках проекта Tofino создается новый интерфейс пользователя для web-браузеров; вместо Gecko используется основанная на движке Chromium платформа Electron, а вместо XUL — web-фреймворк React. Источник: [www.theregister.co.uk](http://www.theregister.co.uk)

➤ Новый проект Civil Infrastructure Platform (CIP) от Linux Foundation — открытая платформа обеспечения функционирования критических элементов гражданской инфраструктуры. Источник: [www.linuxfoundation.org](http://www.linuxfoundation.org)

➤ В первый после открытия исходных текстов проекта официальный выпуск языка программирования Swift 2.2 от Apple добавлена поддержка Linux. Источник: [swift.org](http://swift.org)

➤ Лауреатами учрежденной FSF ежегодной премии "Free Software Awards 2015" стали создатель и основной разработчик GnuPG Вернер Кох и добивающийся превращения библиотек в центры интеллектуальной свободы проект Library Freedom Project. Источник: [www.fsf.org](http://www.fsf.org)

➤ Проект Tor увеличил число альтернативных методов загрузки Tor Browser. Источник: [blog.torproject.org](http://blog.torproject.org)

➤ Переведено на русский язык руководство пользователя LibreOffice Math (исходный англоязычный вариант основан на LibreOffice 4.4, перевод учтывает особенности LibreOffice 5.0). Источник: [forumooo.ru](http://forumooo.ru)

➤ Ссылаясь на «равнодушные пользователей», Google до июля уберет App Launcher из браузера Chrome в версиях для Windows, OS X и Linux, оставив его только в версии для Chrome OS. Источник: [blog.chromium.org](http://blog.chromium.org)

➤ Toshiba отзывает 100 тыс. батарей для ноутбуков в связи с «риском их перегрева и расплавления». Источник: [www.zdnet.com](http://www.zdnet.com)

# 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 межсетевого 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 межсетевого 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.



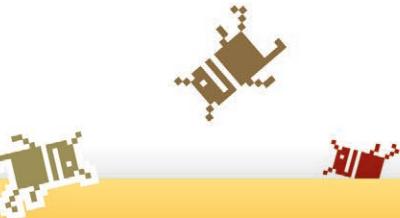
## Назначение устройства

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



# 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,  
eboard, Raspberry Pi,  
Intel Edison, Diligent,  
3D-принтеры  
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)



# Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов



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

## Linux, ZFS и здравый смысл

**В** колонке **LXF205-206** говорилось, что разработчики Ubuntu обещали поддерживать в своем дистрибутиве ZFS. И вскоре в тестируемые сборки 16.04 вошли соответствующие модули. Но это — лишь частичное выполнение обещания: ZFS не поддерживается инсталлятором, и штатного способа установить корневую файловую систему Ubuntu на пул ZFS по-прежнему нет.

Зато почти одновременно вышел дистрибутив, где ZFS взаимодействует поддерживается «из коробки». Это Antergos версии 2016.02.21. В его инсталляторе *Chchi*, после версии 0.14, ZFS можно выбрать наряду с любой другой нативной ФС — правда, только при установке с полного образа и в режиме автоматической разметки целевого носителя.

Но и это не предел желаний и возможностей. В дистрибутиве Proxmox ZFS поддерживается с самой страшной силой: она может не только нести корневую ФС, но и, в отличие от Antergos'a, не требует специального загрузочного раздела с ФС традиционной. Правда, сам по себе Proxmox не очень подходит «для десктопных для целей», однако, будучи клоном Debian'a, может применяться в этом качестве по подключении должных репозиториев.

А тем временем вокруг поддержки ZFS в Ubuntu, даже и столь ограниченной, начинается лицензионная склоки. Инспирированная, как ни парадоксально, не злобными копиастами, а одним из оплотов свободного софта, Software Freedom Conservancy (SFC). Как будут развиваться события — покажет время.  
[alv@posix.ru](mailto:alv@posix.ru)



## Сегодня мы рассматриваем:

### Deepin 15 ..... 14

Дистрибутив с акцентом на дизайн и эстетику, разработанный в КНР, отличается превосходной поддержкой аппаратуры. Правда, требует, чтобы его не запускали в виртуальной среде.

### Solus 1.0 ..... 15

Редкий дистрибутив может похвастаться полной независимостью от каких-либо предшественников, и Solus — один из таких. Посмотрим, стоило ли выпендриваться...

### Da Vinci 1.0 Jr ..... 16

Линейка недорогих принтеров Da Vinci предоставляет технически неподкованным пользователям возможность заняться 3D-печатью,

получив приемлемое качество за вменяемые деньги.

### Acer R11 ..... 17

Этот первый складывающийся хромбук компании Acer (он способен разворачиваться на 360°!) по производительности опередил конкурентов, да и внешностью не обижен.

### Chromebox Tiny ..... 18

Приложите к этому хромбуку от Lenovo монитор, тоже от Lenovo — и получится отличный компактный моноблок.

### Football Manager 2016 ..... 19

Легендарный симулятор футбола — причем не просто матча,



Будь это монитор от Lenovo, Tinyセル бы в док его тыльной стороны.

а и всего того, что за матчами стоит: выбор команды, поиски игроков, трансферы...



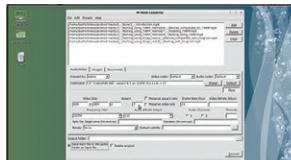
Внутренний голос шепчет, что пейзажик навеян зеленою Ирландией, хотя это вовсе не факт.



Мячик круглый, а поле ровное — но за кулисами идет напряженная организационная работа.

## Сравнение: Видеотранскодеры с. 24

### FF Multi Converter



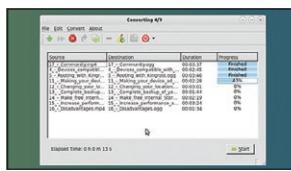
### Handbrake



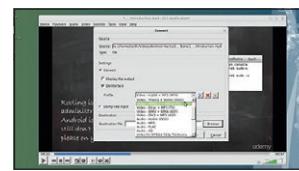
### IFME



### QWinFF



### VLC Media Player



Ситуации, когда требуется перекодирование видео, весьма нередки — и из-за зоопарка кодеков, и из-за зоопарка устройств. Нередки и инструменты для этой цели. Который же из них лучше?

# Deepin 15

Мы не часто сталкиваемся с волнующими и удивляющими дистрибутивами Linux, но **Шашанк Шарма** открыл Deepin и нашел его освежающе непохожим.

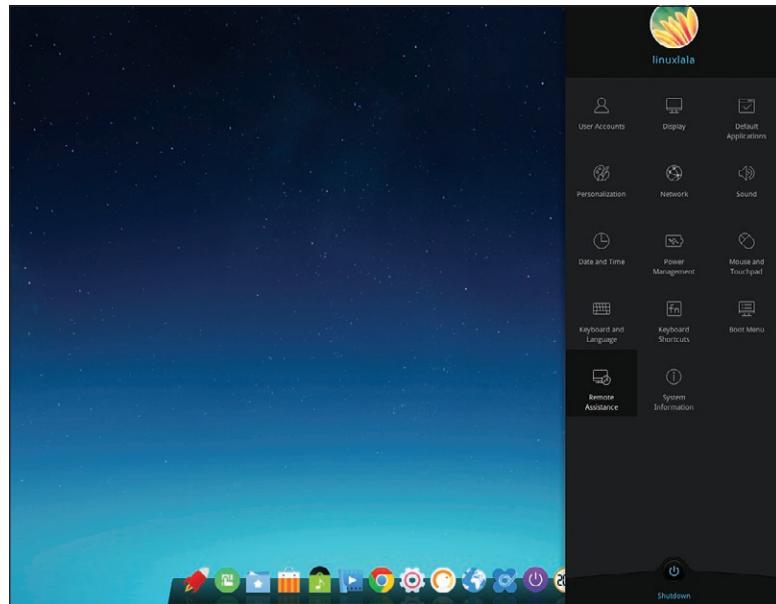
## Вкратце

» В последней версии дистрибутив перешел на Debian в качестве основы. Deepin имеет несколько собственных приложений, включая самый простой установщик всех времен. Дистрибутив серьезно сфокусирован на простоту в использовании и предлагает несколько уникальных функций. См. также: Ubuntu, Netrunner.

Первоначально основанный на Ubuntu, в последней версии Deepin переключился на Debian в качестве своей базы. Несмотря на происхождение, от других производных Ubuntu/Debian дистрибутив отличает собственный рабочий стол под названием Deepin Desktop Environment и самая простая установка, какую мы когда-либо видели в дистрибутиве Linux. Мало того, дистрибутив заходит дальше, представляя себя неординарным дистрибутивом с решением только для установки, для 32- и 64-битных машин. Он также настаивает на том, чтобы его не запускали в виртуальной среде и наслаждались его многочисленными визуально приятными предложениями в полной мере.

Разработанный в Китае, дистрибутив очень заботится об интернационализации, и поэтому установка начинается с просьбы выбрать язык; затем следуют обычные детали, затем установщик автоматически определит ваш часовой пояс и раскладку клавиатуры, но вы можете задать их самостоятельно. Следующий шаг — выбор диска для установки. По умолчанию дистрибутив займет все пространство на выбранном диске, но вы можете вручную настроить разметку, нажав Expert Mode. Это степень участия пользователя в процессе установки. Хотя инсталлятор позволяет выполнять все обычные операции с разделами, Deepin не поддерживает LVM, и вы не можете создать зашифрованную домашнюю папку, что поддерживается в большинстве дистрибутивов.

Три основных компонента дистрибутива — Dock, Launcher и Control Center.



» Deepin первым предлагает весьма утонченный дизайн, и дистрибутив не разочаровывает, доставляя потрясающее визуальное удовольствие.

Launcher хотя и похож на такие аналоги, как Unity и Gnome Shell, но его устройство упорядочено, и он невероятно быстр. При поиске приложений вам стоит только начать вводить буквы — и он предложит все подходящие результаты.

## Плоды усилий любви

Помимо инсталлятора, дистрибутив имеет несколько собственных приложений, таких как Deepin Music и Deepin Store и т.д. Магазин приложений на первый взгляд выглядит как клон Центра приложений Ubuntu, но предлагает гораздо более изящный вариант. Он также предлагает для установки несколько web-приложений, таких как Google Drive, любезно предоставленные Intel Crosswalk Project. Дистрибутив использует репозитории Ubuntu в дополнение к собственным и содержит тысячи полезных пакетов.

В последней версии функция удаления была объединена с лаунчером, а значит, вам не надо запускать магазин или терминал для удаления приложений. Вы можете просто щелкнуть правой кнопкой мыши на приложении и выбрать Uninstall.

Весь дистрибутив разработан с принципом на простоту использования и хорошо подходит для сенсорных устройств. Другое отличие от общепринятого в том, что Control Center интегрирован в рабочий стол,

а не предлагается в качестве отдельного приложения.

Deepin 15 может похвастаться превосходной аппаратной поддержкой, включающей Bluetooth, видеокарты и драйверы принтеров. Control Center обеспечивает доступ к Deepin Manual и Remote Assistance, который позволяет напрямую обратиться за помощью к команде разработчиков. Благодаря акценту на дизайн и эстетику, Deepin является действительно новым ориентиром для дистрибутивов, ориентированных на начинающих пользователей: ему есть чем привлечь взглядел.

### Свойства навскидку

**Features**

- Установка на SSD
- Установка в несколько кликов
- Установка в несколько кликов
- Установка в несколько кликов

#### Свои приложения

Собственные приложения, такие как магазин приложений, в дальнейшем дают пользователю дистрибутива приятные преференции.

#### Умные функции

Интегрирует функцию удаления программ в Launcher и предлагает Control Center как неотъемлемую часть рабочего стола.

## LINUX FORMAT Вердикт

### Deepin 15

**Разработчик:** Wuhan Deepin Technology  
**Сайт:** [www.deepin.org](http://www.deepin.org)  
**Лицензия:** GPL и другие

Функциональность	8/10
Быстродействие	9/10
Удобство в работе	10/10
Документация	7/10

» Созданный удобным для пользователя, дистрибутив идеально подходит начинающим, но продвинутым пользователям предлагает немногого.

**Рейтинг** 8/10

# Solus 1.0

**Шашанк Шарма** тестирует еще один первый релиз дистрибутива, который не хочет умирать, и находит, что проект имеет прочную основу на будущее.

## Вкратце

» Новое дополнение к растущему списку дистрибутивов, предназначенный облегчить неопытным пользователям использование настольного Linux. Дистрибутив использует собственный рабочий стол и утилиты, написанные с нуля, чтобы предоставить знакомую и простую в использовании ОС для новых пользователей. См. также: Pinguy OS, elementary OS.

Постоянные читатели *Linux Format* уже читали о дистрибутивах для начинающих от команды, которая уже выпустила Solus 1.0. Ведущим разработчиком проекта является Айки Доэрти [Ikey Doherty], который ранее работал в Linux Mint Debian Edition. Он уже пытался создавать настольные дистрибутивы для начинающих. Как и в его предыдущих попытках, Solus сфокусирован на рабочем столе и удобстве использования. Solus также не основан на другом дистрибутиве, что дает его разработчикам маневренность в формировании всех аспектов дистрибутива по своему видению, включая пользовательский опыт.

Solus доступен только для 64-битных машин и загружается в live-сессию. Как и большинство основных компонентов, установщик тоже собственный. Хотя установка довольно интуитивна и проста, отсутствие автоматического установщика разработчики должны будут исправить как можно раньше, поскольку просьба к новичкам запустить *GParted* и вручную создать разделы, без всякой помощи, вряд ли их впечатлит.

Создание пользователя выполняется с помощью мастера первой загрузки, который помогает настроить и другие аспекты установки, включая язык, настройки клавиатуры и сети. Мастер также поможет подключиться к Интернету; если это звучит до жути похоже на приложение *gnome-initial-setup*, то потому, что различные аспекты рабочего стола дистрибутива основаны на технологиях Gnome, включая рабочий стол Budgie.



Хотя Solus в настоящее время базируется на Gnome, в ближайшем будущем проект планирует отойти от настольного окружения.

Основанный на GTK, рабочий стол Budgie стремится воспроизвести внешний вид классического Gnome, заодно предлагая некоторые удобства современного рабочего стола Gnome. В меню приложений отображается их список, разбитый по категориям, и есть поле поиска приложений без навигации по меню. Один странный аспект меню заключается в том, что оно реорганизует приложения внутри категорий в зависимости от частоты их использования. Это гарантирует, что часто используемые будут всегда вверху каждого меню, но нужно некоторое время, чтобы к этому привыкнуть. Также нет возможности прикрепить приложения непосредственно из меню приложения. Тем не менее, можно прикрепить на панель открытые приложения и окна.

## Необычные настройки

Другим уникальным аспектом дистрибутива является апплет все-в-одном, центр уведомлений и настроек под названием *Raven*. Он содержит всю информацию в двух вкладках: в одной вкладке отслеживаются все уведомления, а вкладка апплетов отображает календарь, громкость звука и звуковые устройства. Легко настроить рабочий стол с новой темой, как и все аспекты панели. Все элементы на панели, включая меню приложений, часы и т.д., можно размещать и перемещать. По существу, дополнительные аспекты Solus привносят KDE'шные ощущения на рабочий стол.

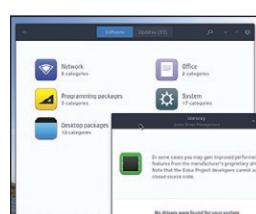
Solus набит приложениями, и за исключением офисного пакета в основном охватывает все необходимое. Если что-либо отсутствует, собственный графический менеджер пакетов и обновлений Solus позволяет обогатить вашу установку, добавив недостающее.

Solus — не просто очередной дружелюбный к новичкам дистрибутив. Это проект с прочной основой и руководством, которое обладает четким видением и достаточным опытом, чтобы его реализовать. Первый релиз делает хороший шаг, и мы полагаем, что через пару релизов дистрибутив должен стать готовым к конкуренции с признанными игроками на равных. **LXF**

**Свойства навески**

### Рабочий стол Budgie

Собственный рабочий стол дистрибутива на базе Gnome — хорошая попытка скрестить старую и новую функциональность.



### Собственные основные утилиты

Все, от менеджера пакетов до утилиты управления драйверами, в этом дистрибутиве написано с нуля.

## LINUX FORMAT Вердикт

### Solus 1.0

**Разработчик:** Solus Project  
**Сайт:** <https://solus-project.com>  
**Лицензия:** GPL и др.

<b>Функциональность</b>	8/10
<b>Быстродействие</b>	8/10
<b>Удобство в работе</b>	8/10
<b>Документация</b>	8/10

» Разработанный с нуля, Solus упорно трудится, чтобы впечатлить не только новых пользователей, но и вообще всех.

**Рейтинг** 8/10

# Da Vinci 1.0 Jr.

**Аластер Дженнингс** очень возбудился, получив невероятно дешевый 3D-принтер, и принялся печатать версию себя — с низким разрешением.

## Спецификации

### » Технология

Метод наплавления нитей

### » Максимальная область печати

15×15×15 см

### » Толщина слоя печати

Тонкий 0,1 мм

Стандартный 0,2 мм

Быстрый 0,3 мм

Скоростной 0,4 мм

### » Печатающая головка

Один экструдер

### » Материал нити

PLA

### » Панель дисплея

2,6" FSTN LCM

### » Аппаратное обеспечение

X86 32/64-бит совместимый ПК с 4 ГБ + DRAM

### » ПО XYZware

### » Тип файла STL, XYZ (3w)

### » Связь USB 2.0, SD-карта

### » Габариты (Ш×Д×В)

42×43×38 см

### » Вес 15 кг

Линейка принтеров Da Vinci предназначена дать технически не-подкованным пользователям недорогой способ заняться 3D-печатью. Принтеры от XYZprinting всегда имели исключительно хорошо выполненные полностью закрыты корпуса профессионального вида, что резко контрастирует даже с более дорогими 3D-принтерами вроде Lulzbot Mini [см. Обзоры, с.18, LXF201], который выглядит как собранный в гараже.

XYZprinting ставит очень ясную цель: построить лучшую машину за самую низкую цену, чтобы охватить наибольшее количество людей. Это означает, что компоненты, такие как подшипники, будут сделаны из латуни или нейлона, а не шариков или полимера.

Jr. — самый маленький и легкий представитель XYZ-принтеров, но поскольку продукты XYZ отнюдь не малы, ему действительно можно называться «юниором» только в сравнении с остальной линейкой XYZ. По сравнению с Ultimaker 2 он выглядит большим, а по сравнению с Ultimaker 2 Go — огромным.

Хотя и маленький, он имеет размеры 42×43×38 см, то есть гораздо меньше, чем первый XYZ-принтер Da Vinci, который имел колосальные размеры 46,8×51×55,8 см.

Качество сборки для XYZ является основным фактором, несмотря на то, что принтер находится в бюджетном ценовом диапазоне. Формованный из пластика корпус и трубчатый внутренний каркас с Т-образными пазами создают ультрапрочную платформу для движущихся частей. Весь принтер полностью закрыт сверху, с боков

и снизу так, что во время печати, пока дверца закрыта, шаловливые пальчики не могут обжечься.

Junior предназначен только для печати из пластика PLA, который поставляется XYZprinting по конкурентной цене, хотя в настоящее время ассортимент цветов и фактур нитей ограничен. Именно из-за печати PLA печатный столик не имеет подогрева и достигает внушительных размеров 15×15×15 см.

После того, как машину достали из коробки, все, что требуется для первой печати — это удаление упаковки и установка катушки нити. Все это займет всего около пяти минут до завершения процесса ввода в эксплуатацию.

## 3D — это просто

Junior имеет холодный столик печати, что ограничивает выбор материала для печати до PLA. Печать основания требует подготовки, чтобы убедиться, что слои основания модели в процессе печати склеиваются. В коробке мы нашли несколько листов текстурированной ленты для столика, приклеенной к стеклянной пластине, которая создает дополнительную липкость, что затем помогает PLA удерживаться на платформе в процессе печати.

Образцы моделей можно скачать с сайта XYZprinting, загрузить на SD-карту с помощью программного обеспечения XYZprinting и использовать программу XYZ Printing, чтобы активировать печать.

Управление программой невероятно простое: обычные ползунки позволяют перемещать модель относительно области сборки перед сохранением модели на SD-карту или прямой печати, если вы подключили кабель USB между принтером и компьютером.

Панель управления может использоваться для навигации по меню принтера для печати с SD-карты. После этого на небольшом ЖК-экране появится список 3D-моделей, для выбора подходящей.

После пяти минут прогрева, необходимого для разогрева рабочего конца нити до нужной температуры, начинается выбранная тестовая печать. На печать потребовалось чуть меньше часа, а полученная модель показала, что это очень способный принтер.



» Мы не уверены, что 3D-принтер Da Vinci выглядит именно так.

Низкое разрешение принтера означает, что края слоев заметны, но простили это, а фактически экструзия дает точные хорошие чистые модели с небольшими признаками обычных проблем (таких как пропуски, наплысы или деформации). Мы также обнаружили, что заводская калибровка Junior не требует никакой перенастройки.

Если вы ищете малобюджетный беспроblemный 3D-принтер, который дает хорошие точные отпечатки, хоть и в низком разрешении, то сейчас действительно нет выбора лучше, чем Da Vinci 1.0 Jr. Простота в использовании и закрытая конструкция принтера корпуса делают его идеальным и дешевым выбором для использования в учебных целях, но если вы хотите более изысканные отпечатки, продолжайте экономить грошики. [LXF](#)

## Свойства навскидку

Features

- Доступ к интернету
- Установка нити
- Установка нити
- Установка нити
- Установка нити

Da Vinci Jr. 1.0 FROM CARD PRINTING

### Простое управление

Благодаря выбору печати прямо с SD-карты или ПК и простому в использовании ПО это идеальный первый принтер.



### Нить

Junior ограничен в выборе материалов, которыми он способен выполнять печать, каковое ограничение — нить PLA.

## LINUX FORMAT Вердикт

### XYZprinting Da Vinci 1.0 Jr.

Производитель: XYZprinting  
Сайт: <http://us.xyzprinting.com>  
Цена: £300

Функциональность	7/10
Производительность	6/10
Удобство в работе	9/10
Оправданность цены	9/10

» Его дизайн и простота использования означает, что это одно из самых серьезных введений в 3D-печать на рынке.

Рейтинг **8/10**

# Acer R11

**Дэнни Торп** днями напролет делает 360-градусное сальто новейшим блестящим хромбуком.

## Спецификации

» ЦПУ 1,6 ГГц четырехъядерный Intel Celeron N3150

» Графика Встроенная Intel HD Graphics

» ОЗУ 2 ГБ (4 ГБ опц.) DDR3L

» Диск 16 ГБ (32 ГБ опц.)

» Дисплей 11,6 HD, 1366×768 LED-IPS сенсорный

» Камера 720р webcam

» Связь 802.11ac (B/G/N) dual-band Wi-Fi, Bluetooth 4.0

» Порты 1×USB 2.0, 1×USB 3.0, 1×HDMI с HDCP, 1×наушники, 1×SD-карта

» Габариты 29×20,3×1,9 см

» Вес 1,25 кг

**В**ыделяясь из группы Chromebook R11 имеет честь быть первым складывающимся хромбуком компании Acer благодаря 360-градусному шарниру. С привлекательным, если не броским дизайном, R11 входит на конкурентную арену бюджетных ноутбуков.

Корпус Acer сделан по большей части из тонкого, белого пластика с матовой отделкой. Однако на крыше машины дизайн отличается: она представляет собой текстурированную белую металлическую панель, которая выглядит прекрасно. Несмотря на то, что R11 по большей части сделан из пластика, в руках он ощущается на удивление прочным.

11,6-дюймовый дисплей имеет разрешение 1366×768 и поддержку сенсорного ввода. На базе R11 сидит слегка утопленная островковая клавиатура с набором слегка текстурированных клавиш. Ниже находится относительно широкий трекпад, достаточно ный для ладони.

Что касается портов, то R11 имеет один порт USB 2.0 и один — USB 3.0, каждый на своей стороне. Есть также полноразмерный порт HDMI, разъем для наушников и слот для SD-карты, чтобы обеспечить перенос файлов.

Машинка оснащена 1,6-ГГц четырехъядерным процессором Intel Celeron N3150 и 2 либо 4 ГБ оперативной памяти. Внутри только 16 ГБ памяти, но вы вряд ли будете хранить много файлов на самой машине. Как и многие хромбуки, он поставляется с 100 ГБ пространства Google Drive на срок два года.



» Экран IPS и шарнир на 360 градусов отлично сочетаются.



» Трекпад — слабое место R11.

В тестах Acer R11 показал себя блестяще, пусть и с небольшими заминками. Сочетание процессора Intel Celeron N3150 и 4 ГБ оперативной памяти удивляет производительностью даже при одновременном вводе текста в Google Doc, воспроизведении без звука пары видео с YouTube и скачивания джемов с Google Play Music.

## Чертовски хорош

В teste Octane Acer выдал результат 8113, опередив Asus Chromebook Flip, у которого — 6795 (чем выше, тем лучше). Аналогично, в Mozilla Kraken Acer R11 опередил Flip, с результатами 4789 и 5447. Дисплей вполне подходит для редактирования документов и просмотра web-страниц. Яркость не составляет ни малейшей проблемы, и углы обзора превосходны благодаря технологии плоскостного переключения (IPS). Кроме того, изображения хорошо выглядели, и движение в нескольких опробованных на ми трейлерах фильмов было довольно четким. Яркой звездой мультимедиа-характеристик R11 является его динамики. Проще говоря, звук на R11 чистый и ясный, и эта крошка умеет быть громкой.

Аcer заявляет срок службы батареи около десяти часов для R11, и мы убедились, что это довольно близко к истине. Мы проводили большинство тестов с яркостью

экрана около 50–60% и нашли, что R11 работает чуть более девяти часов, включая web-серфинг, просмотр отдельных видео и изредка делая записи.

Одной из главных болевых точек R11 является трекпад. Мы просто не в силах примириться с неопределенностью ощущений и неадекватностью нажатия, необходимого для совершения щелчка. Acer Chromebook R11 — хорошая покупка, если вы ищете на рынке нечто способное работать нормально под давлением и предоставить удобство просмотра web-страниц. **LXF**

## LINUX FORMAT Вердикт

### Acer Chromebook R11

Разработчик: Acer  
Сайт: [www.acer.co.uk](http://www.acer.co.uk)  
Цена: £230 (16 ГБ / 2 ГБ)

Функциональность	7/10
Производительность	7/10
Удобство в работе	9/10
Оправданность цены	8/10

» Минималистский внешний вид, обманчиво скрывающий хорошую машину. О, и она может свернуться на 360 градусов в планшет.

**Рейтинг** 8/10

# Lenovo Chromebox Tiny

Модульный ПК с Chrome OS, как решил Чунг Нгуен, понравится и домашним пользователям, и компаниям.

## Спецификации

<b>Процессор</b>	Двухядерный Intel Celeron 3205U, 1,5 ГГц (двуядерный IntelCore i3 5005U, 2 ГГц)
<b>Видеокарта</b>	IntelHD Graphics 5500
<b>ОЗУ</b>	2 ГБ (4 ГБ опц.)
<b>Жесткий диск</b>	16 ГБ
<b>Порты</b>	4xUSB 3.0, Ethernet, HDMI, DisplayPort, комбинированный разъем для микрофона и наушников
<b>Связь</b>	Wi-Fi 802.11n, Bluetooth 4.0
<b>Габариты</b>	17,8 x 18,3 x 3,6 см
<b>Вес</b>	1 кг

С появлением первого форм-фактора микро-десктопа Chromebox Lenovo предоставляет сферам бизнеса и образования возможность создать модульный ПК все-в-одном (AIO) с Chrome OS.

Устройство предлагает такой же компактный плоский форм-фактор, как ThinkCentre M Series Tiny от Lenovo, с Windows, упрощая процесс смены старых офисных Tiny-in-One на новые Chromebox Tiny. Lenovo также отдельно продает 23-дюймовый монитор ThinkVision Tiny-in-One, идущий с 1080р изогнутой нематической панелью со скоростью обновления 5 мс со слотом сзади, куда можно поместить Chromebox Tiny. В данной конфигурации вы по сути создали бы свой собственный модульный моноблок.

ThinkCentre — это компактный настольный ПК, даже более компактный, чем эталонный Apple Mac Mini. Он облачен в прямоугольный черный алюминиевый корпус с пластиковой передней панелью. Снизу у него резиновые ножки для использования в горизонтальном положении, однако можно также поставить Chromebox Tiny вертикально в пластиковый держатель.

Tiny доступен в двух конфигурациях. Базовая конфигурация включает двухядерный процессор 1,5-ГГц Intel Celeron 3205U, тогда как более продвинутая — популярный процессор Intel Core i3 Broadwell. Для каждой модели вы можете выбрать 2- или 4-ГБ ОЗУ. Tiny показал 1683 мс в тесте производительности Mozilla Kraken и 18586 в тесте Google Octane 2.0. То есть это один из самых быстрых имеющихся настольных ПК.

Для сравнения, более мощный процессор Intel Core i5 на ноутбуке Google Chromebook Pixel помог ему выдать немного лучший результат, 23910, в Octane (чем выше результат, тем лучше) и 1430 в Kraken (чем ниже, тем лучше).

Производительность Chromebox была отличной. Мы не заметили никаких задержек при выполнении нескольких задач. Скорость загрузки тоже была высокой: Tiny готов к работе через пять-десять секунд после нажатия кнопки питания. Впечатляет, что Tiny предлагает поддержку 4K, и во время тестов он работал безупречно.

Внутри есть кулер, но мы его вообще не слышали.

Lenovo комплектует Chromebox Tiny проводными USB-мышью и клавиатурой. Мы нашли клавиатуру достаточно удобной, с хорошим ходом, и единственной нашей претензии было то, что она мягковата.

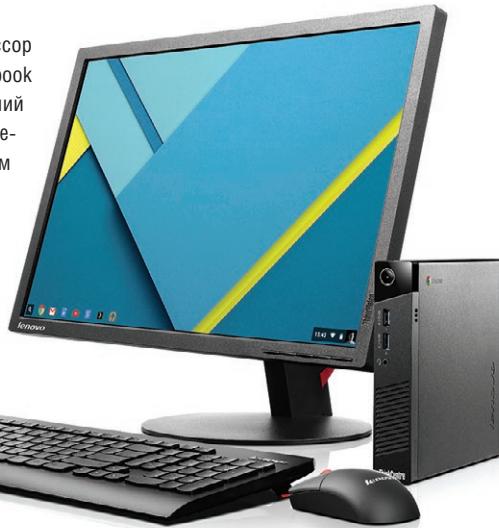
## Модульный дизайн

Lenovo упростили процесс создания десктопа AIO Chrome, привнеся Chrome OS в экосистему Tiny-in-One. Модульный дизайн предлагает пользователям ряд разных преимуществ.

Во-первых, Chromebox Tiny легко обновить. В будущем, если вы захотите при своем экране ThinkVision улучшить производительность самого компьютера, вы просто купите новый Chromebox от Lenovo и пристроите его к задней части монитора.

Во-вторых, если вы захотите перейти от другой ОС к Chrome OS, вы сможете сделать это, отключив Chromebox Tiny от системы ThinkCentre M Series Tiny. И, наконец, в отличие от монолитного десктона все-в-одном, если вы повредите ваш экран или Lenovo в будущем создаст монитор ThinkVision с более высоким разрешением, вы просто сможете поменять монитор и продолжить использовать Chromebox. Экономящий место дизайн Chromebox делает его удобным для использования дома или в офисе, и притом вам не обязательно вкладываться в монитор ThinkVision, если у вас уже имеется другой, пригодный для использования.

В связи с высокотехнологичным оборудованием Chromebox Tiny обеспечивает



Благодаря процессору Intel Core i3, Chromebox Tiny обеспечивает отличную производительность.

отличную производительность. Это один из самых дешевых способов получить процессор Intel Core i3 в настольном ПК; но если вы пойдете по пути Chromebox Tiny, вам придется ограничиться только ОС от Google.

Воспроизведите ли вы песни в Pandora или просматриваете фильм с Netflix, бороздите просторы Сети или проверяете вашу почту, Chromebox Tiny способен справляться с несколькими окнами и вкладками браузера Chrome без замедления работы. Офисные пользователи, выполняющие несколько задач и пользующиеся облаком, найдут в Chromebox Tiny многое привлекательного. **LXF**

## Свойства навскидку



### Core i3

В сочетании с оптимизированной Chrome OS процессор Intel Broadwell работает весьма бойко.



### Разъем монитора

Lenovo предлагает еще и настольный монитор, позволяющий закрепить ваш Tiny на его задней панели.

## LINUX FORMAT Вердикт

### Lenovo Chromebox Tiny

**Разработчик:** Lenovo  
**Сайт:** [www.lenovo.co.uk](http://www.lenovo.co.uk)  
**Цена:** от £160 (2 ГБ)

<b>Функциональность</b>	8/10
<b>Производительность</b>	9/10
<b>Удобство в работе</b>	9/10
<b>Оправданность цены</b>	9/10

» Принесет Chrome OS на десктопы в компактном форм-факторе, позволяя пользователям создать модульный ПК все-в-одном.

**Рейтинг** **9/10**

# Football Manager 2016

Одевшись потеплее, Бен Уилсон готовится засидать бранью и футбольными бутсами якобы дорогостоящих примадонн, притворившихся спортсменами.

## Вкратце

» Популярный симулятор футбольного менеджера на сей раз занялся микроменеджментом. 3D-движок все еще плоховат, а основной интерфейс с точки зрения новичков перегружен. FM включил сомнительную возможность создать свою аватарку, а также отличный аналитический инструмент, хотя и без возможности переносить сохранения из предыдущих релизов.

## Спецификации

Минимальные:  
 » **ОС** Steam OS, Ubuntu 12.04 LTS,  
 » **Процессор** Intel Pentium 4, AMD Athlon 1.8GHz  
 » **ОЗУ** 2 ГБ  
 » **GPU** Nvidia GeForce 7300 GT, AMD Radeon HD 2400, Intel HD 3000, OpenGL 2.0, 256 МБ VRAM  
 » **HDD** 3 ГБ

**Л**егендарный симулятор от Sports Interactive пережил несколько противоречивых сезонов. Критики брюзжат, что он, в своем нынешнем виде, слишком сложен и неповоротлив. Фанаты же говорят — это потому, что он работает, как настоящий менеджер. Критиканы заклеймили его UI как непознаваемый. Новообращенные настаивают, что все необходимое — под рукой, если знать, куда смотреть. Но, по крайней мере, все согласны с тем, что Робби Сэвидж [Robbie Savage] — лакомый кусочек.

Давайте разберемся с этой повторяющейся критикой. Основной интерфейс явно перегружен для новых или давно не практиковавшихся игроков. Там где некогда поиск игрока, выбор команды и характеристики были всего лишь на расстоянии одного щелчка, теперь каждый экран является собой непрерывные столбцы слов, цифр и круговых диаграмм, созданных на основании предположений, исходя из которых, судя по всему, все дороги ведут в тупик. Единственный способ для новичка въехать в игру — взяться за основные задачи, такие как тренировка и обучение управляемого ИИ персонала.

В более старых версиях трансферы и матчи требовали массы времени; теперь это крохотные решения, принимаемые каждые несколько секунд, и это очень сильно нагружает серое вещество и может поглотить все выходные целиком. Выпустите ли вы на поле звездного игрока, вернувшегося после травмы, хотя врач предупреждает, что его хватит не больше чем на 30 минут? Посадите ли посреди матча играющего впол силы защитника на скамейку запасных? Кого вы предскажете победителем на пресс-конференции — QPR или Cardiff — и будет ли другой использовать это как тему для стенгазеты в раздевалке в следующую вашу встречу?



» Основной интерфейс для новых или давно не практиковавшихся игроков перегружен, разве что вы сотрудник Центра полетов НАСА.



» 3D-движок игры имеет привычку разрушать иллюзию реальности FM.

На самом деле, название *Football Manager* вводит в заблуждение; сейчас это больше похоже на *Football Micro-Manager*. И хотя продуманная до мелочей тренировочная жизнь является относительно новой впечатляющей функцией, разработчик никоим образом не утратил понимания, что вывело игру на ее нынешние позиции.

Переговоры о трансферах дьявольски забавны. Согласитесь на цену на игрока, и вам предложат присесть и обсудить с его агентом не только его зарплату, но и дополнительные условия, типа роста рекламных выплат и даже компенсации за простой. Дни матчей тоже удивительно продуманы, с пресс-конференциями, разговорами между игроками и хитроумными тактическими ходами, создающими ощущение, что каждый из них — уникальный случай.

## Football Micro-manager

Спустя семь лет после его появления, 3D-движок все еще не создает настоящего представления о футболе высокого уровня. Он, конечно, вполне приемлем, но для серии, известной своим превосходством во всем, что она делает, «приемлемый» равнозначно «разочаровывающему». Никто не ждет, что он будет выглядеть как FIFA; проблема в том, что хотя бы один раз за игру какой-нибудь игрок да выкинет нечто совершенно абсурдное, разрушив иллюзию реальности.

Вот то, что стало для нас последней каплей: запланированная игра на своем поле с MK Dons. При счете 0:1 моей команде была дана инструкция придерживаться расположения 3-1-3-1-2 и загонять мячи в ворота. На последних секундах центральный защитник Джонатан Вудгейт [Jonathan Woodgate] оказался на правом фланге наравне с зоной пенальти. У него было время пересечь

линию, но он решил отступить. Аж до средней линии. Где и остановился. Тут МК проорвались к воротам и забили завершающий гол. Последовали ругательства и пинки в стену. И мышь шваркнула об пол.

Чем же *FM 2016* хочет завлечь игроков обратно? Во-первых, возможностью создать свой виртуальный образ-аватарку, чтобы бродить по 3D-стадионам. Она ужасна. Плавная интеграция аналитического инструмента Prozone намного лучше. Его включение улучшает происходящее, как на поверхностном уровне — отображение статистических данных из любой точки мира в бегущей строке на экране загрузки — так и на практическом, позволяя изучить любую игру с хирургической точностью.

Игра все еще страдает от упорного отсутствия двух инноваций, созданных ее современником. *Out Of The Park Baseball* давно позволил пользователям не только играть в классических лигах, но и — что крайне важно — переносить свои сохранения из одного релиза в последующие. **LXF**

## LINUX FORMAT Вердикт

### Football Manager 2016

Разработчик: Sports Interactive  
 Сайт: [www.footballmanager.com](http://www.footballmanager.com)  
 Цена: £35

Сюжет	9/10
Графика	6/10
Реиграбельность	10/10
Оправданность цены	7/10

» Отнюдь не игра для мальчуганов, которые загоняют мячи в самодельные ворота: ее глубину оценят даже опытные игроки.

Рейтинг **8/10**



# Мобильные НОВОСТИ

ФРАГМЕНТАЦИИ — НЕТ!

## Samsung для IoT

Южнокорейская компания против фрагментации ОС в мире Интернета вещей.

**K**орпорация Samsung создает встраиваемую операционную систему, надеясь играть большую роль в функционировании миллионов умных бытовых приборов, носимых устройств и промышленного оборудования, составляющих мир IoT. ОС реального времени (RTOS) с открытым кодом, имя которой пока не называется, поможет устройствам быстро выполнять простые задачи без вмешательства пользователя (например, ПО подаст команду на открытие дверного замка и включение освещения при приближении человека к дому). Более подробная информация должна быть обнародована в апреле, в ходе Конференции разработчиков Samsung в Сан-Франциско.

Для RTOS характерна быстрая, практически лишенная задержек обработка данных. Примером такой ОС может послужить VxWorks от Intel,

задействованная в Mars Rover. А открытый исходный код новой ОС увеличивает шансы на ее внедрение и позволяет добиться большей совместимости между устройствами IoT. Samsung пытается предотвратить появление в этом сегменте фрагментации операционных систем, наличие которой среди мобильных устройств создает, особенно для Android, проблемы интероперабельности, совместимости чипов и развертывания обновлений ПО. Не исключено, что новая ОС может оказаться урезанной версией Tizen, уже применяемой Samsung в своих «умных» часах и TV.

Samsung также предлагает платы Artik для «умных» устройств. ОС может позволить разработчикам соединяться с облачной платформой Samsung SAMI, предоставляющей средства аналитики, безопасности и другие сервисы. Возможно, ОС также должна быть совместима с платформами



➤ Стенд на Mobile World Congress, иллюстрирующий концепцию Интернета вещей.

разработки для IoT, типа Google Brillo или ARM mbed. Brillo уже поддерживается платами для разработчиков, в частности, Intel Edison.

РАСШИРЯЕМ ГОРИЗОНТЫ

## SSD на 512 ГБ от Samsung

PM971: заглядываем в будущее мобильных накопителей.

**S**amsung демонстрирует свою точку зрения на то, как выжать максимум возможностей из устройств форм-фактора «смартфон» и «планшет», и с аргументом в виде молниеносных скоростей 1500 МБ/с спорить трудно. На состоявшемся в марте в Японии Samsung SSD Forum компания представила PM971 свой первый SSD с чипом Ball Grid Array (BGA). В отличие от большинства SSD с PGA (Pin Grid Array), контактные выводы BGA представляют собой шарики из припоя, что увеличивает плотность соединений в однокристальном чипе.

Помимо перспективы скорого появления более емких и быстрых накопителей, крошечные размеры форм-фактора BGA позволяют высвободить в корпусе устройства место для более крупной батареи или дополнительных компонентов. С BGA для чипсетов уже работает Intel.

Детальной информации о PM971 мало: пока Samsung просто демонстрирует свои возможности. Однако известно (данные PC Watch), что размерами

PM971 меньше карты SD, т. е. его форм-фактор меньше M.2 (в настоящее время имеются планы относительно стандарта, охватывающего SSD M.2, и неясно, будет ли ему соответствовать предложенная технология, или она новейшая настолько, что просто игнорирует все стандарты). Работает PM971 с новым контроллером Samsung Photon и Flash-памятью MLC V-NAND, и имеет, предположительно, интерфейс PCIe 3.0.

Samsung планирует выпустить три версии, емкостью 128, 256 и 512 ГБ, устанавливая новый стандарт для топовых моделей телефонов, планшетов и гибридных устройств. Обладая последовательной скоростью чтения 1500 МБ/с, PM971 способен предложить последовательную запись на колоссальной скорости 600 МБ/с, что соответствует чтению с произвольным доступом [random read] 190K IOps и записи с произвольным доступом [random write] 150K IOps.

Помимо PM971, на SSD Forum был показан целый ряд новых разработок, включая диск емкостью



➤ Размеры SSD формата BGA позволяют освободить до 10% дополнительного пространства для батареи и уменьшить толщину корпуса мобильного устройства.

4 ТБ для ноутбуков. Похоже, что стагнация в мире мобильных накопителей наконец-то заканчивается и мы увидим дальнейший прогресс смартфонов.

## ANDROID ДЛЯ РАБОТЫ

# ХТ2: ПК на Android

Защищенный ПК от Janam выполнен в виде смартфона для мобильных работников.

**П**роизводитель защищенных компьютеров Janam Technologies представил XT2, устройство весом 280 граммов и с сенсорным экраном 5". Работающее под управлением Android 5.0 (Lollipop) устройство вполне можно назвать защищенным смартфоном, поскольку оно обладает многими функциями последнего, включая поддержку двусторонней голосовой связи; предустановлен браузер Chrome, поддерживаются GSM, GPRS, 4G LTE для передачи данных. Однако производитель называет XT2 не смартфоном, а «защищенным сенсорным компьютером».

Из профессиональных возможностей в нем реализованы сканер Zebra SE4710 1D / 2D Imager и считыватель магнитных карт ANSI / ISO-7813. XT2 соответствует требованиям защиты MIL-STD 810G и электротехнической норме IP67, способен успешно противостоять множественным падениям

с высоты до 1,5 м и погружению в воду на глубину до 0,9 м.

CEO Janam Гарри Лернер [Harry Lerner] уверен, что XT2 «столь же изящен как смартфон... с наиболее передовыми технологиями, удовлетворяющими разнообразные потребности фактически любого мобильного работника». Устройство можно использовать в качестве считывателя RFID и NFC, что, наряду с наличием Wi-Fi, Bluetooth 4.0 и GPS, делает его полностью пригодным для логистических операций. XT2 поставляется с аккумуляторной батареей емкостью 3000 мА·ч, фронтальной камерой 8 Мп и задней камерой 2 Мп, комплектуется 4-ядерным процессором Qualcomm MSM8916 частотой 1,2 ГГц, 2 ГБ ОЗУ и 16 ГБ внутренней памяти, а также слотом micro SD.

Цена зависит от конфигурации устройства и может варьироваться от \$1K до \$2K; при этом,



Janam, представившая XT2, утверждает, что это самый легкий, жесткий и сверхпрочный сенсорный компьютер на рынке.

подчеркивают представители Janam, более высокие объемы сделок значительно снижают цену каждой единицы.

## О ПОЛЬЗЕ КОНКУРЕНЦИИ

# Секретим за так

BlackBerry обновила версию кросс-платформенного мессенджера BBM.

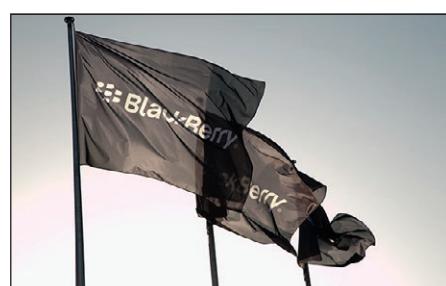
**Р**еализация конкурирующими системами для обмена мгновенными сообщениями функционала конфиденциальности и секретности вынуждает BlackBerry отказаться от абонентской платы за опции "retract" и "timer" в своем кросс-платформенном приложении BBM.

Когда доля BlackBerry на рынке смартфонов снизилась, с целью привлечения новых пользователей канадская компания выпустила версию своего мессенджера BBM для iOS и Android, предоставляя, однако, ряд функций только в качестве платных дополнений. В настоящее время некоторые из этих функций реализованы основными конкурирующими сервисами отправки мгновенных сообщений (например, у Snapchat содержание изображения сообщения исчезают через определенное время), и, чтобы избежать потери пользователей, BlackBerry выпустила обновление, убрав абонплату за опции "retract" и "timer", составлявшую \$0,99 в месяц (это позволяет удалить сообщение или картинку, отправленные из приложения

BBM на телефоны других пользователей (что никак не влияет на уже, возможно, сделанный получателем скриншот с этим сообщением), а также позволяет устанавливать временную задержку, после которой сообщение автоматически исчезнет).

В новой версии BBM добавлена возможность сохранения изображений, полученных с установленных на устройстве приложений. Другие изменения зависят от платформы. Так, в версии для Android отключается звук уведомлений для групповых чатов, реализован просмотр всех опубликованных в чате изображений, добавлена поддержка Marshmallow (Android 6.0). Версия для iOS теперь позволяет записывать и делиться видео большого размера, а также выбирать несколько сообщений для одновременной передачи, редактирования или удаления.

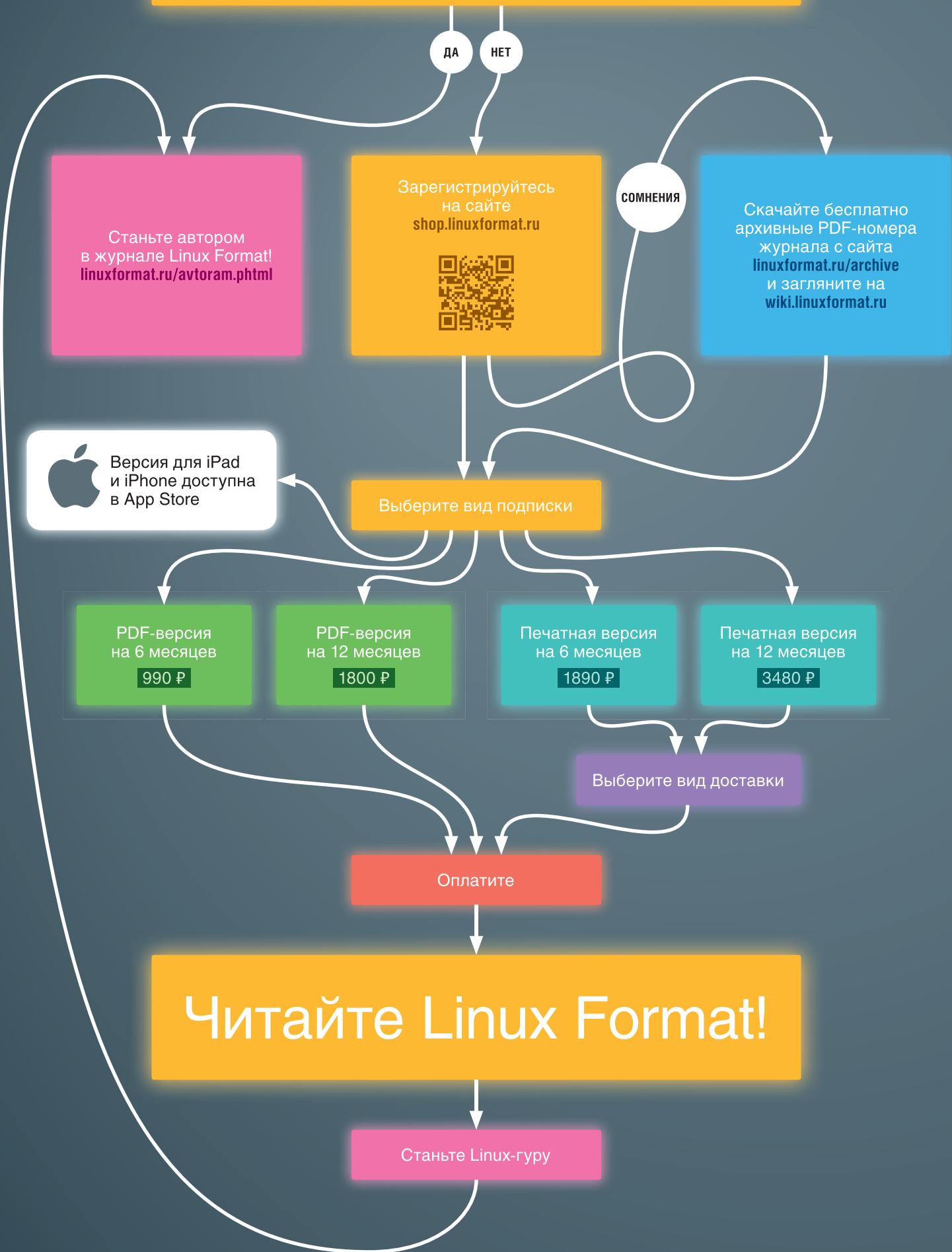
Ряд функций все еще остается платным, включая получение custom PIN или user ID (\$0,99) и удаления рекламы из приложений (\$0,99). BlackBerry предлагает широкий выбор "stickers" — пакетов



BlackBerry больше не взимает дополнительную плату за функции retract и time-delay своего приложения для обмена сообщениями.

для отправки сообщений с изображениями, по цене \$0,99 или \$1,99 (причем подобные дополнения предоставляет за плату не только BlackBerry: с конца 2013 г. Snapchat позволяет пользователям его смотреть идущих сообщений просматривать их во второй раз, а с прошлого года сделал платным повторное воспроизведения). **LXF**

# Знаете всё о Linux?



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

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

# Печатная версия Linux Format понравится читателям, которые:

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

## Способы доставки

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

## Способы оплаты

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



«ГНУ/Линуксцентр»

Санкт-Петербург,  
пр. Медиков, 5, корп. 7  
(метро «Петроградская»)

(812) 309-0686

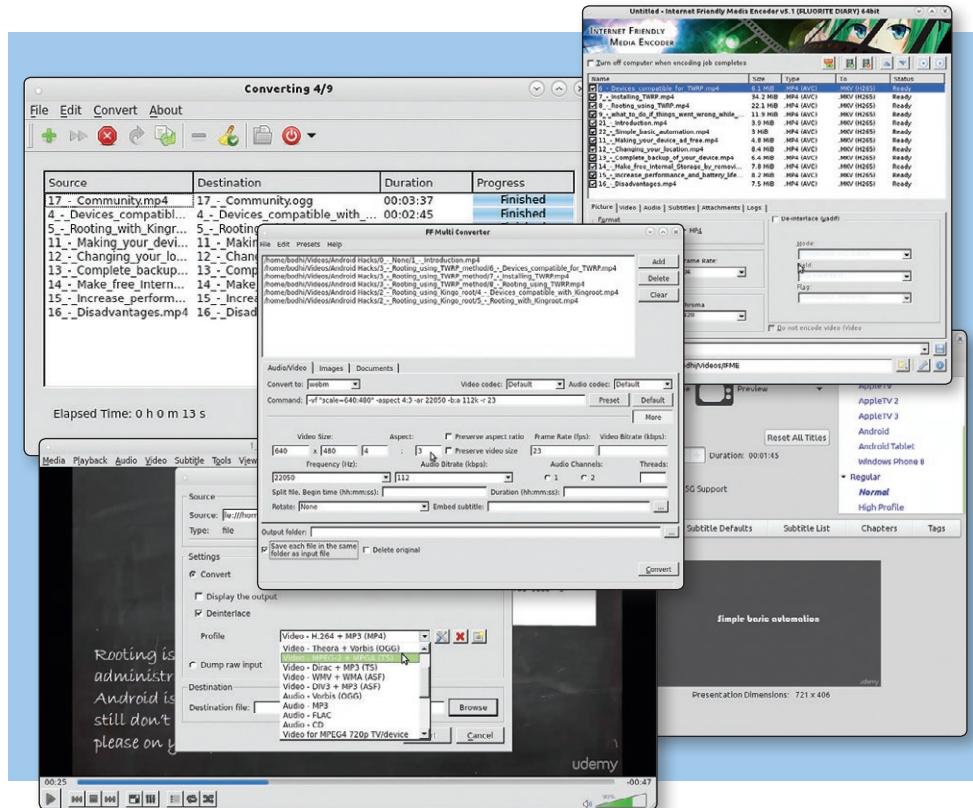
[www.linuxformat.ru](http://www.linuxformat.ru)

# Сравнение

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

## Видеотранскодеры

**Маянк Шарма** с бешеной энергией выискивает лучшую утилиту перекодирования, чтобы смотреть свои мультимедиа на всех устройствах.



### Про наш тест...

Видеотранскодеры предлагают множество функций; обычно они могут конвертировать видео из одних форматов в другие с помощью ряда чередуемых опций. Мы отследим их и составим их рейтинг в соответствии с гибкостью, которые они предлагают в работе. Опытные пользователи ценят множественность настраиваемых параметров, и большинство приложений предлагает ряд преднастроенных опций для быстрого перекодирования видео из одного популярного формата в другой для разных устройств, таких, как планшеты Android и iPad, и т.д.

Эти утилиты основаны на мощных мультимедиа-библиотеках командной строки, а именно *FFmpeg* и *Mencoder*, и мы рассмотрим, как каждая утилита демонстрирует функции движка в графической оболочке: если сумеет при этом не утомлять пользователя — ей фишка. Транскодеры, способные обрабатывать другие источники видео, например, DVD, тоже займут более высокие места.

### Наша подборка

- » FF Multi Converter
- » Handbrake
- » IFME
- » QWInFF
- » VLC Media Player

**В**сё мы таскаем в своих карманах какое-нибудь мультимедиа-устройство. Таковым может являться смартфон, планшет или даже Psion 3. Факт, что сейчас все больше портативных устройств получают достойный экран, и по размеру, и по качеству, и их вполне можно использовать как развлекательные устройства — прямо на ходу. Но тут есть загвоздка. Музыкальные файлы вы можете заказать прямиком в свое устройство без особых проблем, а вот передача видео

с высоким разрешением будет посложнее — в первую очередь потому, что у портативных устройств относительно невелик дисплей. И хотя проникновение многоядерных процессоров на портативные устройства растет и ширится, большинству таких устройств все же маловато мощности для нормального рендеринга видео с высоким разрешением.

В подобных ситуациях надо перекодировать видео из исходной формы в такой формат с таким разрешением, с которыми

сможет адекватно работать портативное устройство. Еще одно популярное применение утилиты перекодирования — конвертирование видео в форматы, подходящие для частных случаев, например, для потокового вещания через сеть или долгосрочного хранения в NAS. В нашем Сравнении мы рассмотрим самые популярные утилиты перекодирования видео и поможем вам выбрать ту, что предлагает наибольшее количество функций в простом для работы интерфейсе.

# Опции настройки

На какую гибкость они способны?

**Э**то одна из незначительного числа областей, где ни одно из тестируемых приложений нас не разочарует. Все выбранные приложения богаты настраиваемыми опциями и предлагают достаточно широкие возможности повлиять на перекодируемое видео. Например, все они позволяют выбрать любой из поддерживаемых контейнеров, видео- и аудио-кодеков для перекодируемого видео. Более того, используя любой из транскодеров, легко повлиять на отношение сторон кадра, отдельно указать частоту кадров и битрейт и даже встроить файлы субтитров.

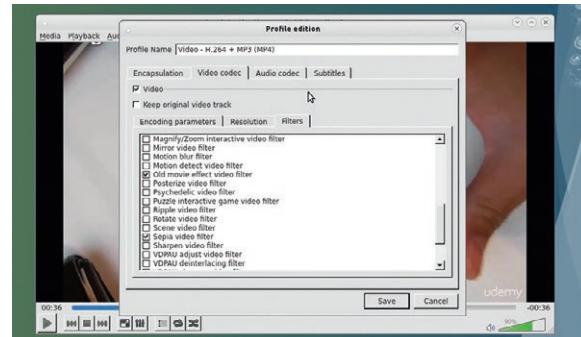
*FF Multi Converter* заодно помогает быстро настроить задачу перекодирования, предоставляя опцию сохранения перекодированного файла в той же папке, что и исходный файл, и удалить оригинал после перекодирования. Можно также велеть этому приложению поделить полученное видео на части заданной длительности.

Подобно *FF Multi Converter*, *Handbrake* имеет отдельные вкладки для указания настроек Picture, Video и Audio. *Handbrake*

вдобавок предлагает выпадающий список оптимизированных настроек для обычных типов исходных материалов, а именно фильмов, анимации и изображений. Как и с видео, *Handbrake* умеет менять битрейт и частоту сэмплирования для программы кодирования аудио.

*QWinFF* еще богаче опциями для работы с аудио. Можно вручную настроить громкость и даже изменять скорость. Транскодер умеет конвертировать все видео- и аудиоформаты, поддерживаемые *FFmpeg*. Более того, *QWinFF* позволяет перекодировать только фрагмент видео, указав время начала и окончания, конвертировать несколько файлов в один формат вывода одновременно или выбрать разные форматы вывода для каждого файла ввода.

Даже *VLC*, который вообще не является специализированным транскодером, как все остальные в нашем Сравнении, предлагает впечатляющий набор настраиваемых параметров. К тому же в медиа-плейере вы сможете добавить дополнительный аудиотрек в видео. Выдающаяся функция



» Набор видеофильмов *VLC* позволяет преобразовать видео совершенно новым способом.

*VLC* — умение прилагать к видео все виды фильтров: например, придавать видео вид старого фильма; более теплый тон с эффектом сепии; увеличить контраст через увеличение резкости видео; и более десятка других эффектов.

Аналогично, можно использовать *VLC* для настройки параметров кодирования для аудио и выбрать из этого десятка эффектов, например, задержку звука, нормализатор громкости, и т. д.

Наименьший выбор предлагает *IFME*. Он поддерживает только кодек H.265 и умеет перекодировать видео только в форматы MKV или MP4.

## Вердикт

<b>FF Multi Converter</b>
★★★★★
<b>Handbrake</b>
★★★★★
<b>QWinFF</b>
★★★★★
<b>VLC</b>
★★★★★
<b>IFME</b>
★★★★★

» *IFME* лишился большая за ограничение количества поддерживаемых форматов всего двумя.

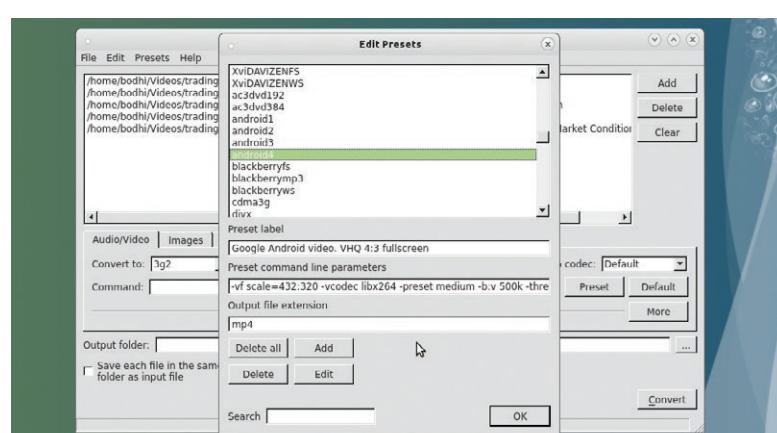
# Готовые профили

Перекодирование для новичков.

**В**се приложения предлагают несколько предварительно заданных настроек, оптимизированных для определенных устройств или случаев применения, и скоренько перекодировать видео с ними очень удобно.

У *IFME* шесть преднастроек-пресетов для сохранения перекодированного видео для воспроизведения на Android, Windows и в Интернете с разной частотой кадров, плюс пять пресетов определяется пользователем. Однако приложение не предлагает опции сохранения индивидуальных настроек.

*QWinFF* тоже не позволяет создавать и сохранять индивидуальные настройки. Приложение берет набор своих настроек в проекте *WinFF*, и они отображаются в зависимости от выбранного формата контейнера. Так, если вы хотите конвертировать видео в AVI, *QWinFF* перечислит настройки для создания видео, которое будет, среди прочих платформ, воспроизводиться на Nokia N810 или Creative Zen, а выбор MP4 отобразит настройки для разных устройств, таких, как iPod, SanDisk Sansa и т. д.



» *FF Multi Converter* позволяет просматривать и редактировать параметры командной строки всех своих настроек.

## Вердикт

<b>FF Multi Converter</b>
★★★★★
<b>VLC</b>
★★★★★
<b>Handbrake</b>
★★★★★
<b>QWinFF</b>
★★★★★
<b>IFME</b>
★★★★★

» Список преднастроек *Handbrake* длинен, но покороче списка, предлагаемого *FF Multi Converter*.

# Дайте порулись

Трудно ли управляться с каждым из видеотранскодеров?

**Э**ти графические инструменты получают всю свою мощь и функции от богатых функциями и мощных утилит командной строки, но разработчикам надо постараться не просто впихнуть в них добавочные функции, чтобы утереть нос конкурентам. Подобная тактика, разумеется, позволяет опередить их по отдельно

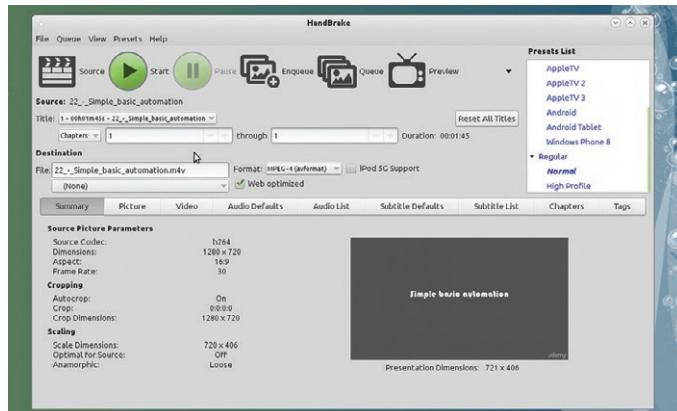
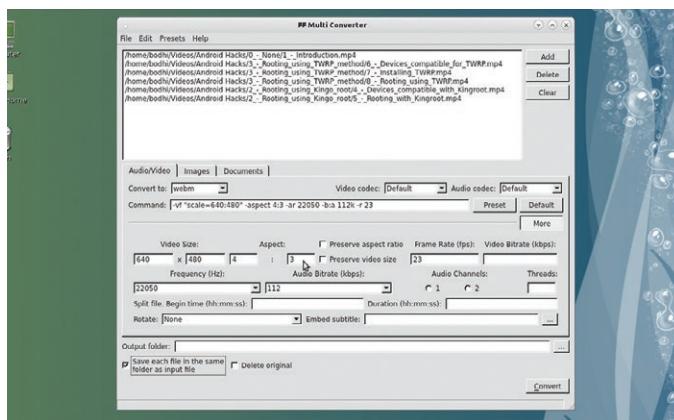
оцениваемым параметрам в конкретной номинации, но в конечном итоге приведет к проигрышу лучше организованному инструменту.

Учитывая природу нашей задачи, мы не рассчитываем отыскать транскодер с минимальным интерфейсом, но попытаемся обнаружить транскодер с интерфейсом понятным. Мы будем высматривать

интуитивные приложения, предлагающие не только достаточный объем контроля, который окажется разумен и посильен для неопытных пользователей, но и расширенные возможности для тех, кто глубоко разбирается в процессе перекодирования и способен самостоятельно выбирать параметры для желаемых файлов.

## FF Multi Converter ★★★★☆

У этого приложения простой интерфейс с большой областью вверху, куда добавляются файлы для перекодировки. Нижняя часть интерфейса состоит из трех вкладок: Audio/Video, Images и Documents; все три предлагают четкие опции для каждой задачи. Процесс перекодирования включает выбор видеофайла и затем выбор нового формата файла. Для выбранного формата файла автоматически выбирается видео- и аудиокодек по умолчанию, но у вас есть и опция выбрать любой другой из имеющихся. Новые пользователи могут нажать на кнопку Preset [Преднастройка] и выбрать одну из перечисленных опций перекодирования видео. Но в основном *FF Multi Converter* является графической оболочкой для *FFmpeg*, а значит, опытные пользователи могут сделать куда больше. Кнопка с надписью More [Еще] предоставит им доступ к таким продвинутым опциям, как изменение объема файла, битрейт и встроенные субтитры, и т.д.



## Handbrake ★★★★☆

По сравнению с другими, интерфейс *Handbrake* весьма загруженный, и на первый взгляд способен запутать. Да, здесь множество кнопок и перечислено несколько настроек, но те, которые в данный момент не используются, окрашены серым. Как и в других приложениях, процесс перекодирования начинается с указания *Handbrake* на видео, которое надо конвертировать. Но, в отличие от других, вы также можете установить диск DVD или Blu-ray и перекодировать его содержимое. Указав исходное видео, вы можете задать конвертирование под конкретное устройство, выбрав его среди предварительных настроек. Опытные пользователи также могут сами подправить расширенные настройки. Интерфейс *Handbrake* предлагает удобные подсказки по инструментам, появляющиеся при наведении на них мыши, благодаря чему его расширенные настройки более доступными даже для неопытных пользователей.

# Документация и поддержка

Рука помощи для новичков.

**X**отя все видео транскодеры в нашем Сравнении более или менее прости в использовании, вам, вероятно, понадобится заглянуть в их документацию, прежде чем пробовать их расширенные функции. У *VLC* минимальный интерфейс для функции перекодирования, и документация такая же. Этой теме посвящена всего одна страница на *wiki* проекта, и она скорее про перекодирование из командной строки, чем про навигацию по графическому интерфейсу. *VLC* часто отсылает пользователей к своей функции

потокового вещания, которая подобна перекодированию, и эта тенденция отмечается также и на официальных форумах.

*FF Multi Converter* ушел недалеко: его аскетичная *wiki* перечисляет имеющиеся преднастройки и предлагает краткое руководство по установке приложения из исходника и на Ubuntu. Да и *IFME* предлагает в своей *wiki* лишь минимум информации: 7 страниц по установке, созданию плагинов и введению в перекодировщики AAC.

Аналогичный шаблон распространяется на *QWinFF*; его документация предлагает

беглое введение в применение инструмента с краткими данными по его использованию, и имеется в разных форматах для онлайн- и офлайн-потребления. Притом у *QWinFF* нет форумов, *wiki*, IRC или списка рассылок.

Инфраструктура поддержки *Handbrake*, в полную противоположность остальным, включает очень активные форумы сообщества, канал IRC и *wiki* с полезнейшим разделом ЧаВо. Имеется также очень подробное руководство пользователя, которое объясняет все аспекты приложения.

## Вердикт

### Handbrake

★★★★☆

### FF Multi Converter

★★★★☆

### IFME

★★★★☆

### QWinFF

★★★★☆

### VLC

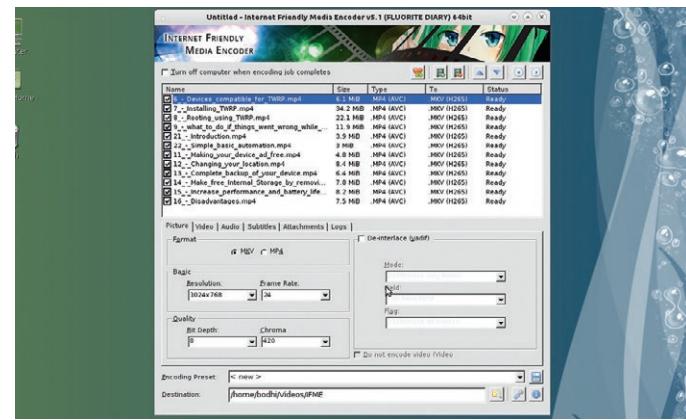
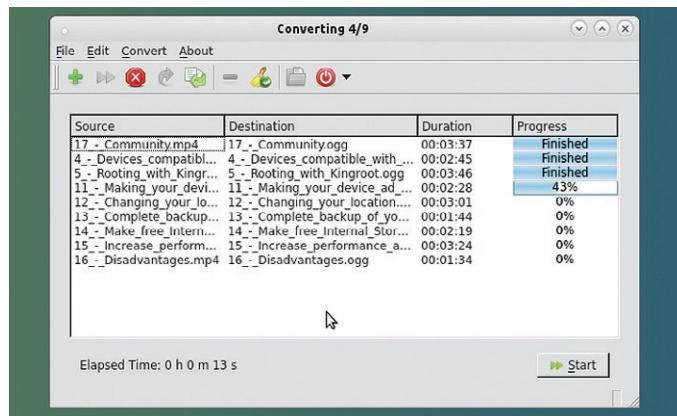
★★★★☆

» *Handbrake* выигрывает благодаря своему сообществу — самому заботливому из всех.

**IFME** ★★★★☆

*IFME* — одно из более ресурсоемких приложений, и требует Mono. Простой интерфейс *IFME* не слишком интуитивен: вы начинаете с размещения видео в верхней части приложения, но затем предполагается, что вы переместитесь в среднюю часть интерфейса и затем перейдете вниз, чтобы выбрать преднастройку для работы с настройками в средней части. Вы можете или начать перекодировать файлы прямо из интерфейса, или сохранить очередь и вызвать *IFME* из командной строки, чтобы задействовать для процесса перекодирования максимум ресурсов.

Опытные пользователи могут прервать преднастройки и вносить изменения в разные аспекты процесса перекодирования, такие, как аудио и видео; это делается через разные вкладки. В отличие от ряда других приложений, *IFME* не предоставляет предпросмотра видео перед его перекодированием со встроенными настройками.

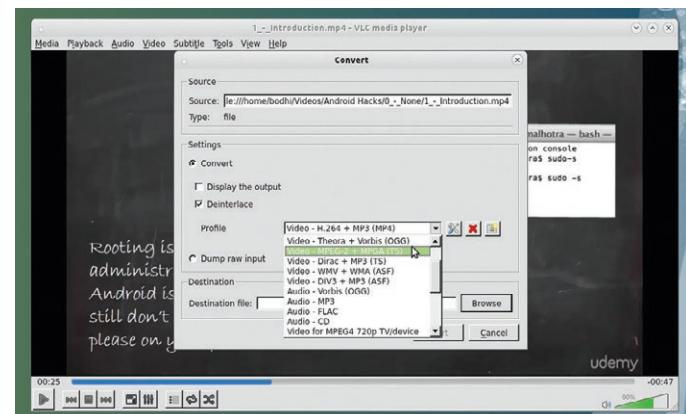
**QWinFF** ★★★★☆

Интерфейс *QWinFF* минимален, но интуитивен и приятен в работе. Вместо демонстрации всех имеющихся опций, *QWinFF* использует двухступенчатый мастер. Вы начинаете с добавления файлов и выбора формата файлов, после чего отобразится список подобающих преднастроек. Для более тонкой настройки нажмите на кнопку Edit [Редактировать]. Перед вами появится еще одно окно, где вы сможете изменить различные параметры конвертирования. Помимо очевидных вкладок Audio и Video, имеется еще вкладка Time [Время], которая позволяет графически выделить и осуществить предпросмотр части видео, которую вы хотите перекодировать. И, наконец, есть вкладка Advanced, где перечислены опции командной строки *FFmpeg*, соответствующие выбранным вами настройкам.

Опытные пользователи могут изменить их вручную и передать в настоящую команду, которая и выполняет перекодирование.

**VLC** ★★★★☆

Популярный плейер, как и *QWinFF*, использует для перекодирования посредственный мастер. Однако его подход не столь аккуратен или логичен, как у *QWinFF*. Процесс начинается с закачки файлов для перекодирования, и затем при желании закидывается файла с субтитрами. Использование кнопки More [Больше опций] позволяет просмотреть опции для добавления к перекодируемому видео аудиотрека. Затем вы нажимаете на кнопку Convert/Save [Конвертировать/Сохранить], чтобы перейти в следующее меню. Однако в отличие от традиционного мастера назад вернуться нельзя, и если вы забыли добавить музыкальный трек и захотите сделать это позднее, придется выйти из мастера и начать процесс заново. На следующем экране выбирается один из готовых профилей. Здесь опытные пользователи могут использовать кнопку Edit для выбора аудио- и видеокодеков и изменения параметров перекодирования.



# Поддерживаемые форматы

И что они тут перекодируют?

**K**ак и следовало ожидать, поскольку *IFME* способен перекодировать только видео с помощью кодека H.265, в данном тесте он плеется позади всех. У этого приложения также имеется ограниченная поддержка контейнеров всего с двумя опциями — MKV (по умолчанию) и MP4.

И *QWinFF*, и *VLC* поддерживают множество всяких форматов благодаря своему движку FFmpeg. Он поддерживает многочисленные кодеки и широкий диапазон видео- (MPEG1/2/4, DivX, WMV, Theora и т.д.)

и аудиоформатов (AC3, AAC, FLAC, MP2/3 и т.д.). Тем не менее, и *QWinFF*, и *VLC* ограничивают свои задачи конвертированием видео.

Следующим будет *Handbrake*, с поддержкой видеокодеков H.264, MPEG-4, MPEG-2, VP8 и Theora. В отличие от других наших приложений, *Handbrake* способен перекодировать мультимедиа, полученные из разных источников, например, с DVD и дисков Blu-ray, если у них не предусмотрено защиты от копирования, и это важное преимущество.

В пику другим инструментам нашего Сравнения, *FF Multi Converter*, оправдывая свое название, помимо аудио и видео, лихо конвертирует документы в разнообразных популярных форматах [Ред.: — А это у нас по теме?], включая DOC в ODT, XLS в ODS, ODT в PDF и т.д. Более того, *FF Multi Converter* справляется даже с конвертированием изображений, заодно выполняя ограниченные задачи по их редактированию — например, обрезать, повернуть, перевернуть, и справляется со многими форматами, например, BMP, PNG, JPG и не только.

## Вердикт

### FF Multi Converter

★★★★★

### VLC

★★★★★

### Handbrake

★★★★★

### QWinFF

★★★★★

### IFME

★★★★★

■ *FF Multi Converter* также справляется с конвертированием изображений и документов.

# Поддержка командной строки

Для убеленных виртуальными сединами.

**K**ак и с остальными задачами, пройдя достаточно долгий путь в перекодировании, вы, возможно, захотите немного сэкономить времени и конвертировать видео через командную строку. Некоторые приложения, например, *QWinFF* и *FF Multi Converter*, по сути являются всего лишь графическими оболочками мощных библиотек командной строки, которые не предлагают собственного интерфейса.

*IFME* называет себя «гибридом GUI-CLI», и вы можете использовать графический

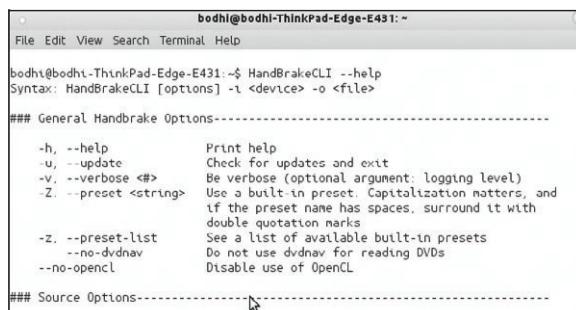
интерфейс для настройки задач перекодирования и сохранить очередь в файле XML. Затем можно выйти из приложения и попросить его обработать сохраненную очередь из командной строки, не вызывая графического интерфейса и экономя тем самым ценные системные ресурсы. Учтите, однако, что об этом нет практически никакой информации на сайте проекта, и опция `--help` в интерфейсе командной строки вас тоже особенно не порадует: информации она дает мало.

Зато у *VLC* в его *wiki* масса информации по перекодированию из командной строки. Практически все задачи по перекодированию, выполняемые через графический интерфейс, можно выполнить из интерфейса командной строки *VLC*: например, сменить аудио- и видеокодеки, битрейт, разрешение видео, обрезать видео с любой стороны или применить фильтры из CLI. Опция **-H** познакомит вас со всеми подробностями использования.

## Вердикт

**FF Multi Converter**  
★★★★★  
**Handbrake**  
★★★★★  
**QWinFF**  
★★★★★  
**VLC**  
★★★★★

» Желая рабо-  
тать с команд-  
ной строкой,  
не пожалейте  
времени  
на освоение  
FFmpeg.



➤ *HandbrakeCLI* — это отдельная от графического приложения утилита.

# **Расширенные функции управления**

Что предлагается искушенным пользователям?

**X** отя наши приложения можно использовать и не имея опыта кодирования видео, практически во всех есть некоторые функции управления для ветеранов в этой области. Некоторые приложения основаны на мощных библиотеках командной строки, и у опытных пользователей есть шансы повозиться с добавочными функциями управления, порой способными расширить полномочия графического интерфейса.

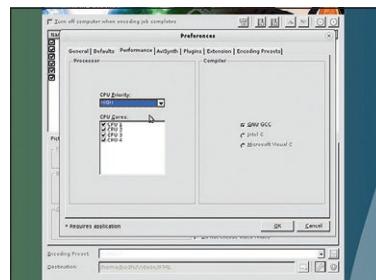
*QWinFF* предлагает меньше возможностей по расширенным опциям, чем некоторые другие, но позволяет ввести дополнительные параметры во все инструменты командной строки, участвующие в процессе перекодирования, в т. ч. *FFmpeg*, *ffplay*, *SoX* и т. д. *VLC* тоже не отличается большим объемом контроля над использованием ресурсов. Его расширенные опции ограничиваются окошком для устранения чересстрочной развертки видео и возможностью создавать индивидуальные профили, выбрав формат контейнера соответствующих аудио- и видеокодеков.

Они вдвоем являются собой контраст *IFME*, который предлагает несколько опций

для опытных пользователей, а также впечатляющий контроль за потреблением ресурсов. По умолчанию он задействует при перекодировании все наличные ядра CPU, но вы можете опционально попросить его ограничиться только частью из них, оставив другие для выполнения прочих задач.

*IFME* умеет не только влиять на видео самыми разными способами, но и кодировать видео в несколько проходов, чтобы сохранить наилучшее качество. *IFME* также способен кодировать и сжимать аудио без потерь, применяя ускорение OpenCL; правда, подробная информация о том, как это делается, отсутствует. Кроме того, *IFME*, один из немногих, позволяет выбрать кодировщики аудио и задавать их настройки, например, битрейт, частоту и количество каналов для кодирования.

И еще есть *FF Multi Converter*, который позволяет вводить дополнительные параметры командной строки при выполнении функций редактирования видео и даже определять индивидуальные команды FFmpeg. В *Handbrake* есть отдельная вкладка для редактирования изображений и аудио, и он позволяет присоединять к видео тэги



➤ Можно контролировать, как *IFME* потребляет CPU, выбрав одну из шести преднастроенных уровней CPU Priority: от Real-time [В реальном времени] и High [Высокого] до Normal [Обычного] и Low [Низкого].

метаданных, например, Title [Название], Actors [Актеры], Directors [Режиссеры], Plot [Сюжет], Release Date [Дата Выхода], Genre [Жанр] и т.д. Интерфейс *Handbrake* включает несколько настроек, имеющих смысл для опытных пользователей: например, опцию варьирования частоты кадров или двухпроходного кодирования. Вы можете также использовать вкладку Video, чтобы передать в энкодер дополнительные настройки.

## Вердикт

**IFME**  
★★★★★  
**FF Multi Converter**  
★★★★★  
**Handbrake**  
★★★★★  
**QWinFF**  
★★★★★  
**VLC**  
★★★★★  
» Стратегия Handbrake приводить расширенные опции в главном интерфейсе может путать новичков.

# Лучшие транскодеры

# Вердикт

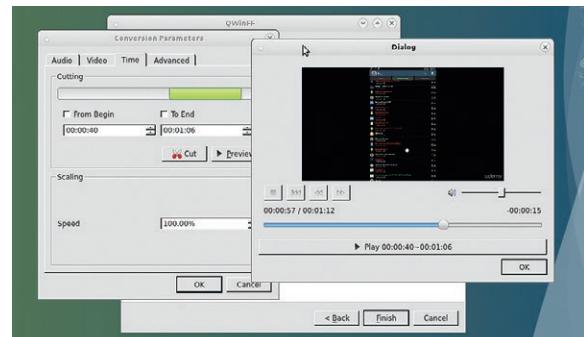
**Э**то одно из тех Сравнений, где выбрать победителя довольно трудно. Дело в том, что найдется область применения для каждого из них. Однако нашим победителем будет тот, что полезен большинству пользователей. Подчеркнем, что у каждого из приложений есть свой конек. Например, *FFmpeg* поддерживает меньше всех кодеков — зато является, вероятно, наилучшей опцией для создания оптимизированного видео для потокового вещания. Гибридное использование им GUI и командной строки означает удобство группового перекодирования видео с одинаковыми настройками, и быстее, чем другие приложения.

Подобным же образом *FF Multi Converter* является хорошей графической оболочкой для *FFmpeg*. Это приложение предлагает множество функций управления и поддерживает массу форматов. Увы, в нем нет опции, позволяющей просматривать файл перед его перекодированием, и хотя он умеет делить файлы на части,

но не предусматривает механизма сделать эти изменения видимыми. Однако если вы хотите конвертировать не только видео, то это приложение отлично справляется с конвертированием изображений и документов в разные популярные форматы.

У нас был большой соблазн объявить победителем данного Сравнения *VLC*. В конце концов, это один из лучших мультимедиа-плееров, и функции перекодирования, предлагаемые *VLC*, вполне конкурентоспособны. Нас также впечатлило наличие видео- и аудиофайльтров, которые позволяют внести своеобразные штрихи в кодируемое видео. Но хотя у нас не вызывает нареканий его употребление в качестве многоцелевого медиа-плеера, ему не хватает некоторых функций, обязанных быть в специализированном транскодере: например, предпросмотра файлов и визуального отображения вносимых изменений.

Единственная причина, по которой в победители не вышел *Handbrake* — его ограничения в поддержке контейнеров и кодеков.



➤ **QWinFF** — более привлекательный эстетически и обернутый в Qt родич *WinFF*, от которого и берет список своих преднастроек.

Если вам надо перекодировать видео для оффлайн-воспроизведения на своих устройствах, вы можете дать *Handbrake* шанс, и его удобные подсказки помогут вам осуществлять навигацию по его довольно загруженному интерфейсу.

Но если вам нужен транскодер видео, который будет простым и с легкостью справится с работой, возьмите *QWinFF*. У него мощная сердцевина и симпатичные дополнения — например, предпросмотр и опция визуально настраивать видео, чтобы помочь вам с удобством выполнять работу.

**QWinFF прост и с легкостью справляется с работой.**

I

**QWinFF** ★★★★☆Версия: 0.2.1 Сайт: <http://qwinff.github.io> Лицензия: GPL v3

» Удобен для новичков и имеет достаточно опций для опытных пользователей.

IV

**FF Multi Converter** ★★★★☆Версия: 1.7.2 Сайт: <http://bit.ly/FFMultiConverter> Лицензия: GPL v3

» Верное своему имени, приложение конвертирует также изображения и документы.

II

**Handbrake** ★★★★☆Версия: 0.10.2 Сайт: <https://handbrake.fr> Лицензия: GPL v2

» Лучшая опция перекодирования видео для любого оффлайн-использования.

V

**IFME** ★★★★☆Версия: 5.1 Сайт: <https://x265.github.io> Лицензия: GPL v2

» Одна из лучших опций для оптимизированного видео при онлайн-стриминге.

III

**VLC** ★★★★☆Версия: 2.2.1 Сайт: [www.videolan.org/vlc](http://www.videolan.org/vlc) Лицензия: GPL v2

» Идеален для всех, кому нужно урезать количество используемых приложений.

## Обратная связь

Регулярно перекодируете видео? Согласны с нами или, по-вашему, *Handbrake* лучше? Или применяете нечто иное? Пишите нам: [lxf.letters@futurenet.com](mailto:lxf.letters@futurenet.com).

## Рассмотрите также...

**В**ас не устроил наш список? В репозиториях вашего дистрибутива есть множество иных опций. Одна из первых, на которую стоит взглянуть — *WinFF*, с очень простым интерфейсом. Еще один транскодер на базе *FFmpeg* — *FFmpeg YAG*. Этот проект до сих пор имеет отметку альфа, и его единственный релиз, похоже, не предлагает

ничего выдающегося и отличного от конкурентов, но на него стоит взглянуть. Транскодер *Cutter* также основан на *FFmpeg* и довольно популярен. Однако его последний стабильный релиз был в 2014 г., а последняя версия 2.0 находится в стадии бета с середины прошлого года. Есть еще *MakeMKV* — в основном это приложение Windows,

но сообщество пользователей портировало его и создало версию Linux. Если вам нужно нарезать DVD, может также попробовать *OGMRip* в качестве альтернативы *Handbrake*. И, наконец, есть такие приложения, как *Avidemux*, умеющие перекодировать видео в порядке дополнения к их основной работе видеоредактора. **LXF**

# ТАЙНЫ ХАКЕРОВ

Путешествуем с **Джонни Бидвеллом**  
по растущей стране  
 злоумышленников и изучаем  
 их атаки — с целью самозащиты.

CONFIDENTIAL



**K**ак бы нам ни хотелось верить в обратное, но Интернет — это не одно сплошное благолепие, радуга и единороги. И не то что сообщества радуг и единорогов онлайн не процветают — они-то процветают, но заодно не бедствуют и подлые хакеры, жаждущие похитить, выпытать, испортить, заразить или повредить любые цифровые ценности, до которых способны дотянуться их загребущие лапы.

Ваши вычислительные мощности, сайты, пароли, списки контактов и, конечно, информация о кредитных картах — всё это предметы алчности интернет-мерзавцев, стремящихся любыми средствами заполучить такие ресурсы. Иногда это бывают простые приемы социальной инженерии — фишинговые сообщения по электронной почте, замаскированные под сообщения от вашего банка; или нечто более тонкое,

например, вредоносный код JavaScript, встроенный в стороннее рекламное объявление на вашем любимом сайте. Результаты могут быть любыми — от неудобства (например, необходимости проходить через раздражающую длинную процедуру регистрации пароля) до неприятностей (потери персональных данных, например, фотографий) и полной катастрофы (опустошения вашего банковского счета или кражи вашей идентификационной информации). Однако хакерские атаки довольно редко направлены на конкретного пользователя, то есть обычно, как говорится, «ничего личного»: они, скорее, целятся на определенные сайты или некоторую инфраструктуру.

**Атаки редко направлены против конкретного пользователя: обычно «ничего личного».**

фий) и полной катастрофы (опустошения вашего банковского счета или кражи вашей идентификационной информации). Однако хакерские атаки довольно редко направлены на конкретного пользователя, то есть обычно, как говорится, «ничего личного»: они, скорее, целятся на определенные сайты или некоторую инфраструктуру.

И снова, мотивация в основном корыстная, но могут быть и иные цели: например, промышленный шпионаж или искажение вида сайта. Все становится куда серьезнее, если вовлекается государство. При подобном сценарии потенциальные жертвы могут обнаружить, что их оборудование повреждено еще до того, как оно успело покинуть

производство, или что многие узлы Тор настроены таким образом, что происходит атака, направленная на устранение анонимности, или что они перенаправлены на поддельную версию их любимого проверен-

ного сайта, которая теперь будет собирать идентификационную информацию. Работники Belgacom пали жертвами последней — так называемой атаки Watering Hole — являясь частью программы Quantum Insert от GCHQ, которая перенаправляла их на содержащие вредоносное ПО версии LinkedIn и Slashdot.

# Защита настольных ПК

Следуйте этим советам дома — и вы не угодите в печальную статистику облапошенных.

**Б**езусловно, есть вирусы Linux, и безусловно, есть антивирусные программы для Linux, но в целом в них нет особой нужды, разве что для применения самыми параноидальными домашними пользователями — в основном благодаря тому, что к скачиванию нежелательных программ в мире Linux относятся неодобрительно. Предпочтительный способ работы — через ваш менеджер пакетов для скачивания программ; он проверяет целостность и аутентичность всего скачиваемого, и вам нечего волноваться при установке Ask Toolbar [Ред.: — Однако!]. Фактически, менеджер пакетов, или, вернее, его частое применение (вместе с разумной политикой резервного копирования) — это лучший способ охраны здоровья вашего Linux.

Основные дистрибутивы быстро выпускают заплатки к обнаруженным уязвимостям системы безопасности, и вы должны сразу их применять. Если вы не используете дистрибутив на исходниках, типа Gentoo, обновление пакетов займет считанные секунды (ну, может, несколько секунд, если вы давно этого не делали), так что у вас нет причин отнекиваться. Более ориентированные на пользователя дистрибутивы даже напомнят вам об этом сами, из своего графического интерфейса. Если у вас или у вашей тетушки Этель стоит доисторическая установка системы, побалуйте себя и обновите эту систему. Велики шансы, что если оборудование работало со старой Ubuntu, оно ничуть не хуже — а возможно, даже лучше — сработается с новой. А если у тетушки Этель — один из миллиона компьютеров, на которых по-прежнему работает Windows XP, то совершите разумный поступок и установите Linux.

Вследствие стандартов Интернета (а в случае Flash их склонны игнорировать), многие атаки являются кросс-платформенными. Поэтому уязвимость в Chrome (или Chromium) можно обнаружить и использовать так, что в равной мере пострадают пользователи Windows, OS X и Linux. Подобным же образом следующая уязвимость Flash (вероятно, она появится через неделю после того, как вы это прочитаете) может затронуть всех пользователей. Легко насмехаться над Flash и высмеивать ужасный выбор его авторов в плане программирования, однако правда в том, что негативные отзывы о нем объясняются его былой популярностью. К счастью, Интернет немного повзрослев, и мы осознали, что в конечном итоге проприетарные плагины — не самая удачная идея. Даже Adobe и Microsoft согласились с этим, рекомендуя своим пользователям отказаться от Flash и Silverlight. И кто мы такие, чтобы спорить? Есть еще несколько чудаковатых сайтов, которые используют Flash,

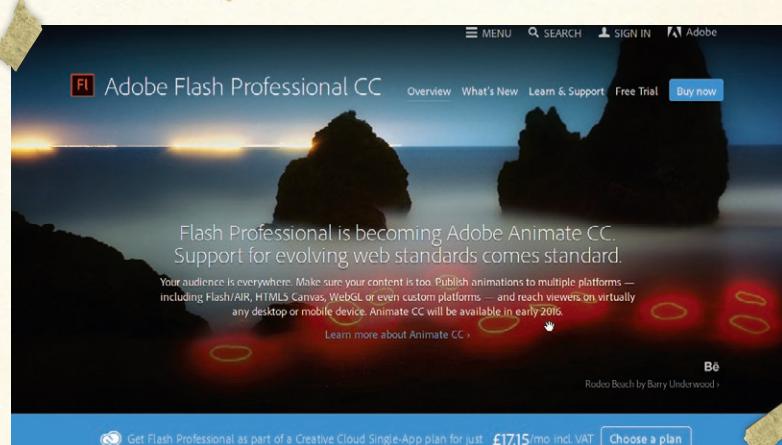
но большинство (включая YouTube) для всех своих виджетов перешло на HTML5.

Так почему бы не посмотреть, проживете ли вы без Flash? Для начала отключите его в браузере, и если спустя несколько дней блуждания по Интернету вы ровно ничего не заметите (кроме отсутствия раздражающей рекламы), то можете смело удалить негодника. А если вы заходите на сайты, где он по-прежнему нужен, обдумайте установку плагина вроде flashblock, чтобы аппараты Flash воспроизводились только при нажатии.

Немало также и вредоносного JavaScript, так что, блуждая по темным закоулкам Сети, не забудьте использовать плагин вроде NoScript, чтобы его остановить. Вероятно, вам придется внести некоторые надежные сайты в белый список, потому что большая часть Сети без JavaScript не работает. Все больше сайтов громко осуждают блокировщики рекламы. А мы заявляем, что пока рекламные сети уязвимы для злоумышленников, а сайты используют назойливую и вредную рекламу, пользователь должен иметь право на спокойную жизнь в Сети.

И, наконец, помните о необходимости обновлять прошивку своего роутера: инфицированному роутеру легко нанести ущерб хорошо настроенному компьютеру, соединенному с ним, и многие сетевые атаки нацелены как раз на эти устройства. Иногда об этом заботится ваш интернет-провайдер, но вы, возможно, решите заменить оборудование своего провайдера на роутер с чудесным DD-WRT [см. стр. 72, LXF198].

➤ Flash, ax, a-ax...  
Даже Adobe пыта-  
ется дистанциро-  
ваться от своего  
кросс-платформен-  
ного плагина с по-  
мощью хитроумно-  
го ребрендинга.



## Интернет (плохих) Вещей

Очевидно, что будущее наступает — и не менее очевидно, что оно подразумевает подключение вашего холодильника или утюга к Интернету. У нас нет ни малейших сомнений в том, что весьма удобно, когда ваш холодильник сам заказывает в магазине молоко, решив, что его запас вышел, или когда отопление включается само, узнав о вашем выходе с работы, или когда можно автоматизировать некоторые задачи — например, полить собаку или покормить цветы. К сожалению, технология, вызвавшая всё это к жизни, в то же время

является катастрофой, которая затаилась в ожидании своего звездного часа.

Все это будет основываться на объединенных в сеть встроенных системах, которые по большей части не дают возможности поставить заплатки: да, на большинстве из них окажется Linux для ARM или MIPS, но большинство оборудования будет применять двоичные драйверы. И поэтому при обнаружении уязвимостей в системе безопасности (а они будут обнаружены) вы не сможете пропатчить эти устройства. Пользователи, желающие избежать

инфицирования, должны будут их отключить (сделать свой дом менее умным) или приобрести модель поновее. Отчасти спасает Raspberry Pi, где есть такая замечательная штука, как GPIO для соединения со всеми Вещами (сенсорами, камерами, вашим сердцем), и уж это-то устройство вы полностью контролируете. Обычно дополнения к Raspberry Pi идут с драйверами с открытым кодом, и открытая природа сообщества означает, что драйверы должны содержаться в полном порядке до далёкого будущего.



# Методы хакинга

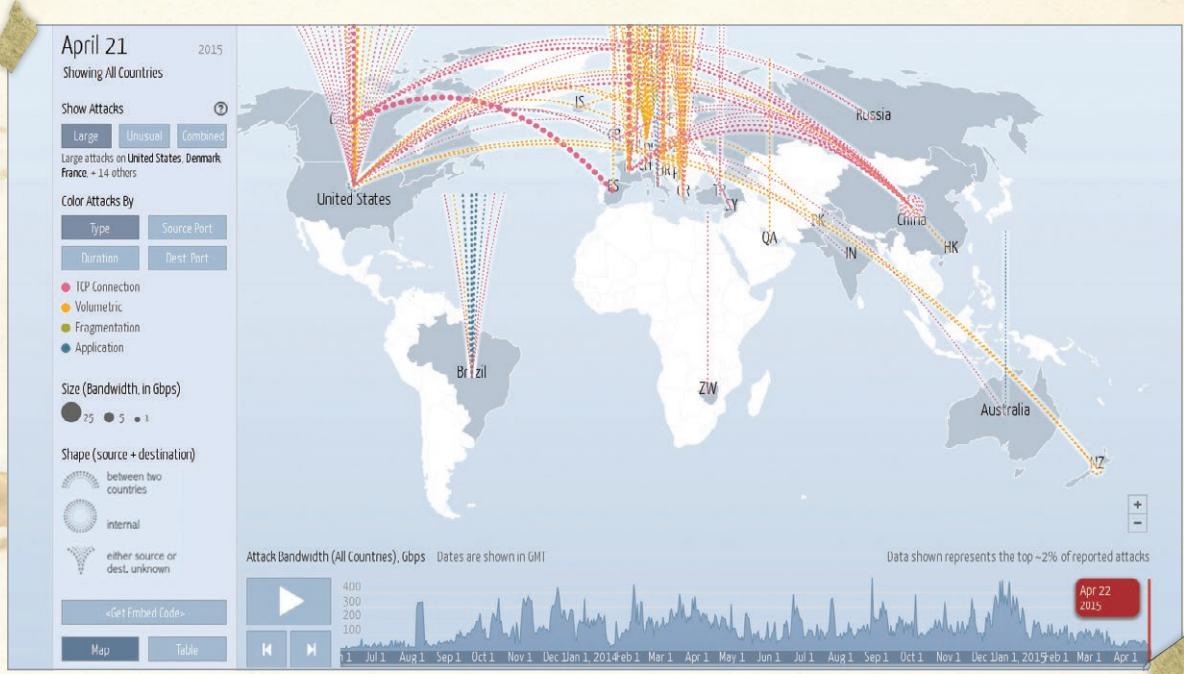
Узнав о трюках злоумышленников, незамедлительно спланируйте свою защиту.

**П**режде чем щебетать обо всех чудных и загадочных способах компрометации данных, мы начнем с самого общего: социальной инженерии. Однозначно, самый простой способ для злоумышленника узнать ваш пароль — просто спросить его, или заставить вас ввести его на сайте, который находится под чужим контролем. Урок, который вы должны извлечь из этого, заключается в обычной осторожности: то, что с виду сообщение пришло от вашего знакомого, вовсе не означает, что так оно и есть. Такие сообщения даже бывают отправлены с той самой учетной записи, но это может оказаться всего лишь следствием взлома.

Если оно всего лишь содержит ссылку на некий мутный домен с не относящимся к делу текстом, то почти наверняка не является подлинным, и вам не стоит нажимать на эту ссылку. Подобным же образом мошенники могут отправлять сообщения электронной почты, имитирующие сообщения вашего банка. Они часто предлагают «обновить настройки безопасности», предлагая ссылку на сайт, очень похожий на сайт вашего банка. Они часто регистрируют похожего

вида доменные имена, например, заменяя 1s на ls (посмотрите, как здорово это работает со шрифтом **LXF**). К счастью, в последние годы все больше применяются так называемые сертификаты Extended Validation (EV) SSL. Они подвергаются тщательнейшему изучению Certificate Authority, и если они подтверждены, то ваш браузер отобразит название компании в веселой зеленой строке рядом с веселым зеленым замочком. Ваш банк почти наверняка использует сертификаты EV, поэтому убедитесь, что сертификат проверен в вашем браузере, прежде чем вводить любые сведения. Выясните также политику своего банка насчет общения по электронной почте; многие банки указывают в сообщениях последние четыре цифры номера вашего счета, и при получении сообщения якобы от своего банка без этой информации вам стоит насторожиться.

С конца 1990-х существуют атаки SQL-инъекций, и, как многое из того времени [Ред.: — Например, чувство стиля нашего научного редактора], они необъяснимым образом продолжают существовать, ничуть не интересуясь развитием. На самом деле



Tem, кто не верит, что DDoS-атаки реальны, [www.digitalattackmap.com](http://www.digitalattackmap.com) докажет в реальном времени, что они не правы.

## DDoS — кувалда сетевых атак

Distributed Denial of Service (DDoS, распределенный отказ в обслуживании) — это, вероятно, наименее изящный метод вышибить сервис из Сети. И один из популярнейших. По сути это бомбардировка сайта трафиком со множества разных хостов (отсюда — распределенный), чтобы он оказался не в силах обслуживать законные запросы или вообще рухнул.

Есть инструменты, например, Slow Loris и Low Orbit Ion Cannon (LOIC), которые можно употребить,

чтобы направить всю мощь своего канала на выбранные вами хосты, возможно, объединив свои усилия с единомышленниками; однако мы настоятельно не рекомендуем так делать, потому что это противозаконно. Большинство DDoS-атак сейчас проводятся ордами скомпрометированных компьютеров, которые именуются botnet.

DDoS-атаки руководствуются простой математикой: запросы нужны короткие (чтобы отправлять их в большом количестве), но способные оказать

мощное воздействие на сервер — или в силу ресурсоемкости их обработки (например, захлебывания CPU и памяти жертвы), или потому, что ответы на них требуют большой пропускной способности канала (занимают исходящую сеть хоста). Перед самым Рождеством 2015 г. BBC вышибли в оффлайн рекордной атакой на 602 ГБ/с. Беспокойство вызывает то, что группа, взявшая на себя ответственность за эту атаку, заявила, что они «просто провели свои возможности».

это не совсем так, есть некоторые вариации вокруг основной темы; но главная идея в том, что код SQL так или иначе втишиается на компьютер и либо выдает информацию о базе данных, либо заставляет эту базу данных плясать под дудку злоумышленника. Классические примеры обычно связаны с тщательно отобранный пользовательской информацией из web-формы. Например, нередко в качестве интерфейса к базе данных используется PHP. Предположим, у нас есть простая форма, которая просит пользователя ввести имя пользователя и пароль и использует такой фрагмент PHP для проверки этой информации (через стандартный запрос HTTP POST) в базе данных MySQL:

```
$query = "SELECT * FROM users WHERE username = '" + $username +
        "' AND password = '" + $password + "'";
```

На вид — разумно: мы старательно вложили одиночные кавычки, чтобы переменные PHP нормально передались базе данных. И было бы разумным, кабы мы подчищали введенную пользователем информацию. (Предположим, что переменные \$username и \$password переданы из нашей формы в первозданном виде). Проблема — в нашем убеждении, что наши пользователи не будут передавать через нашу форму необычную информацию. А вдруг в поле пароля ввели zelda' OR 1=1? Обратите внимание на незамкнутую кавычку: она запутывает запрос SQL, заставляя его считать, что всё после кавычки — часть самого запроса. Итак, даже если мы введем недействительное имя пользователя или имя настоящего пользователя, чьим паролем не является Zelda, запрос всё равно будет соответствовать всем пользователям благодаря трюизму 1=1. В некоторых движках SQL в систему в результате входит первый пользователь из списка, а обычно это администратор. Иногда требуется приложить чуть больше усилий, чтобы этот трюк сработал — полученный в результате запрос не является действующим SQL из-за несовпадающих кавычек, однако это можно обойти, например, добавив в конце вредоносного фрагмента комментарии.

SQL-инъекции бывают и более продвинутые/разрушительные, чем вышеописанная: например, можно ввести команду (DROP TABLES), которая скоренько всё удалит из активной базы данных, или добавить в базу данных своего собственного пользователя, обеспечив себе черный ход на сайт. Для ввода вредоносной информации нам даже не нужна форма: мы можем сделать это, манипулируя URL или даже создав собственный запрос HTTP POST. Обобщенно это переход из контекста данных (который мы в качестве пользователя контролируем) в контекст запроса (где нашего присутствия вообще не предполагается).

Атаки SQL-инъекций происходят что ни день, но с ними проще всего справиться, как мы увидим далее. Как ни удивительно, жертвой подобной атаки в 2015 г. пала TalkTalk (британская телекоммуникационная компания). Изначально их пресс-релиз сообщил о некой «последовательностной атаке», однако в конце концов комментаторы выяснили, что компания пала жертвой этой самой тривидальной из всех типов атак. Персональная информация сотен тысяч пользователей угодила в руки злоумышленников, и на данный момент арестованы и выпущены под залог пятеро подростков. Пост Троя Ханта [Troy Hunt] «Все, что вы хотели узнать о внедрении SQL (но боялись спросить)» (<http://bit.ly/TroyHuntUOnSQLInjections>) определенно стоит прочитать, если вы хотите узнать больше о SQL.

Межсайтовый скриптинг [Cross-site Scripting] (XSS) — название, данное классу брешей, характерных тем, что пользователь может внедрить собственные фрагменты кода или скрипты на сервер, при неспособности сервера их распознать. Внедренный код обычно написан на JavaScript, поэтому выполняется на стороне клиента. На самом деле это не прямая атака на сервер, а скорее атака на других пользователей на нем.

```
Edit View Search Terminal Help
[jonni@jbmachine ~]$ cd sqlmapproject-sqlmap-c37f485/
[jonni@jbmachine sqlmapproject-sqlmap-c37f485]$ sqlmap.py sqlmap.conf sqlmap.py
[jonni@jbmachine sqlmapproject-sqlmap-c37f485]$ python2 sqlmap.py -u https://linuxformat.com/forums/index.php?id=1
[1.0-dev-nongit-20160112-0b43]
http://sqlmap.org

[*] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers as well as users and operators must assume full responsibility for any misuse or damage caused by this program

[*] starting at 15:46:48

[15:46:53] [INFO] testing connection to the target URL
[15:46:54] [INFO] checking if the target is protected by some kind of WAF/TPS/TDS
[15:46:54] [WARNING] reflective value(s) found and filtering out
[15:46:54] [INFO] testing if the target URL is stable
[15:46:55] [WARNING] target URL is not stable. sqlmap will base the page comparison on a sequence matcher. If no dynamic nor injectable parameters are detected, or in case of junk results, refer to user's manual paragraph 'Page comparison' and provide a string or regular expression to match on how do you want to proceed? [(C)ontinue/(S)top/(R)eject/(Q)uit]
[15:47:07] [INFO] searching for dynamic content
[15:47:08] [INFO] dynamic content marked for removal (81 regions)
```

Скрипт Python *sqlmap* умеет сканировать на предмет любых уязвимостей к SQL-инъекциям, но обязательно прочтите предупреждение и следуйте ему.

## INXS(S)

Эти типы атак играют на том доверии, которое люди испытывают к сайтам синдикации (XSS), поскольку редко склонны питать подозрение к знакомому доменному имени, которому они уже, вероятно, разрешили (если они используют NoScript или Adblock или нечто подобное) выполнять скрипты. Код можно внедрить в URL, или через комментарий либо пост на форуме. Ранним примером является червь Samy, который в 2005 г. распространился по MySpace (тогдашней основной социальной сети). Сам по себе червь был довольно безобиден — он просто добавлял в профиля некий текст, вынуждая своих жертв стать друзьями автора, Сами

## АТАКИ SQL-ИНЪЕКЦИЙ ПРОИСХОДЯТ ЧТО НИ ДЕНЬ, НО С НИМИ ПРОЩЕ ВСЕГО СПРАВИТЬСЯ.

Камкара [Samy Kamkar]. Увы, все профили, которые он затрагивал, оказывались зараженными, так что за считанные часы Сами обзавелся более чем миллионом друзей. MySpace почти сразу прикрыл его учетную запись и разобрался с проблемой, однако свидетельства того происшествия до сих пор можно найти среди реликтов MySpace. Сначала для Сами все было довольно грустно — до него добрались агенты секретных служб и на три года запретили ему пользоваться компьютером, однако он не сдался и стал ведущим исследователем проблем безопасности (загляните на его сайт: <http://samyl.pl>).

Мы много слышим в прессе о похищении баз данных в результате взлома сервера. Злоумышленнику, который так или иначе умудрился спрятать базу данных, еще придется потрудиться, чтобы извлечь из нее пользу. Помимо информации по кредитным картам (которая, надо надеяться, всё же не хранится в полном объеме), есть еще одна вещь, чрезвычайно интересная для злоумышленников, и это — пароли. К счастью, любое приложение, которое не зря ест свой хлеб с солью (синтаксической), хранит хэш с синтаксической солью, а не сами пароли в виде простого текста. Если у двух пользователей одинаковый пароль, синтаксическая соль обеспечивает наличие разных сохраненных хешей. Обнаружить хэш-конфликт (для безопасных функций хэширования, типа SHA256) трудно, но такие инструменты для атаки грубой силы, как *John the Ripper*, могут автоматизировать этот процесс.





# Защищать и охранять

Подключенный к Сети (или просто сети) компьютер с Linux несет особую ответственность.



**Б**ольшинство дистрибутивов предлагают по умолчанию весьма разумные настройки безопасности, однако простор для улучшений всегда есть. По мере включения новых сервисов появляются и новые векторы для атаки, и поэтому вам стоит проверять, какие сервисы вы используете — или через *top*, или запуска `$ systemctl status` на системах с *Systemd*. Отключите все, что вам не нужно, и изучайте то, чего не понимаете.

Самая популярная конфигурация — пакет LAMP (или его эквивалент), который требует только запуска *Apache* и *MySQL*. Конечно, вам надо будет также запустить SSH, иначе вы не сможете заниматься администрированием.

И вот что гораздо важнее для сисадминов: обновляйте свои пакеты! Люди кидаются на новые уязвимости, как только о них сообщают, а награды за первенство среди жертв нового вредоносного кода нет. Если вы на самом деле хотите обезопасить себя, следите за новыми уязвимостями (например, на сайте Common Vulnerabilities and Exposures details, <http://www.cvedetails.com>). Если появится нечто способное затронуть вас, а ваш дистрибутив еще не обновился до пропатченной версии, вам, вероятно, надо будет отключить соответствующий сервис, пока не подготовят новые пакеты. Нередко крайности в этом нет — отчет об уязвимости может содержать временные обходные пути (например, изменения настройки), которые можно употребить, чтобы сервис продолжал работать.

## Обезопасим Secure Shell

Есть пара вещей, которые настраиваются легко, а вашу безопасность увеличат очень резко. Начнем с блокировки сервиса SSH. Неплохая идея — не разрешать удаленный вход в учетную запись root; если нужен доступ root, можно сперва войти как непривилегированный пользователь и использовать sudo или su. Убедитесь, что вашему пользователю разрешены эти команды, иначе при добавлении вами строки

```
PermitRootLogin no
```

к `/etc/ssh/sshd_config` и перезапуске сервиса прав root вы не получите. Но зато и никто другой не получит. Стоит также отключить пароли логинов, используя вместо них для аутентификации ключи SSH. Вы можете сгенерировать пару ключей на своей локальной машине и затем загрузить публичный ключ на свой сервер, сделав так (соответствующим образом заменив пользователя и хост на свои и нажав Enter, чтобы принять размещение ключа по умолчанию):

```
$ ssh-keygen
```

```
$ ssh-copy-id user@host
```

Первая команда генерирует пару 2048-битных RSA ключей, а вторая приложит публичный ключ к файлу `~/.ssh/authorized_keys` на вашем сервере. Теперь вы должны быть в состоянии войти через SSH, указав свой приватный ключ:

```
$ ssh -i ~/.ssh/id_rsa user@host
```

Позаботьтесь о файле, содержащем ваш приватный ключ; некоторые любят хранить его на USB-флешке. Естественно, вам нельзя ее терять или оставлять валяться где-нибудь в общественном месте. Убедившись, что всё это работает, и обретя уверенность в своих способностях управлять ключами, можете отключить логин и пароль, добавив

```
PasswordAuthentication no
```

к `/etc/ssh/sshd_config` на сервере. Пароли логинов по определению небезопасны, люди используют их неоднократно, они слишком коротки и очень часто основаны на словарных словах. Любая учетная запись пользователя на вашем сервере, имеющем доступ суперпользователя, должна быть защищена ключом. Стоит настроить fail2Ban, чтобы клиенты, предпринявшие несколько неудачных попыток входа в систему, временно блокировались. На <http://bit.ly/Fail2BanOnDebian7> есть отличное руководство на эту тему — оно написано для Debian 7, однако прекрасно подходит практически для любого Linux.



➤ Metasploit Framework (есть на нашем диске) — ценный ресурс для тех, кто занимается тестированием на вторжения, с чем даже эта ASCII-корова согласна.

## С ветераном такого быть не должно

К сожалению, как бы тщательно вы ни планировали свою защиту, вероятность пасть жертвой хакеров все равно присутствует. Возможно, вы почуствуете неладное из-за того, что исказился внешний вид вашего сайта, или из-за того, что вам больше на него не войти (через SSH или любую панель сетевого управления). Возможно, вам позвонит обозленный хозяин сервера, обвиняя вас в соучастии по DDoS-атакам. Если вы объясните ему ситуацию, возможно, он проявит понимание и поможет. А если вам всё-таки удастся войти, то вы должны,

в порядке обязательной меры предосторожности, отключить все сервисы.

Далее вам нужно будет скачать журналы вашей системы, web-сервера и базы данных и извести немало чашек чая, знакомясь с ними. При известном везении вы сможете выяснить точное время взлома. Возможно, у вас есть резервные копии, но если вы не уверены в том, что они были сделаны до атаки, не стоит их использовать для восстановления. Если вы предоставляете VPS, то переустановка — дело нехитрое, поэтому подумайте

о восстановлении своего сайта именно таким образом. Старые базы данных и файлы настройки можно с успехом восстановить, однако сначала проверьте их. И, наконец, предупредите всех пользователей вашего сайта о вторжении в систему. Вероятно, эта новость их не обрадует, но они должны быть в курсе. Если у вас хранятся важные данные, то по британскому законодательству вы также обязаны уведомить о проникновении в вашу систему еще и Управление комиссариата по информации [Information Commissioners Office].

## Укрепляйте свой web-сервер

Обслуживание статических web-страниц, например, из *Apache*, установленного на *Debian* с настройками по умолчанию, вряд ли причинит вам какие-то неприятности. Но всегда есть место для улучшений. *Let's Encrypt* (<https://letsencrypt.org>) отныне предлагает бесплатные и простые в использовании сертификаты SSL, и пользователи теперь могут прескокойно бродить по вашему сайту, зная, что это и вправду ваш сайт и что трафик шифруется. Если на своем сайте вы работаете с важными данными (например, личной или финансовой информацией), вам, вероятно, надо будет оплатить как минимум проверенный на доменном уровне сертификат.

Однако мы не решаем проблемы, наляпывая на них сертификаты: мы можем также настроить шифр, чтобы не применять уязвимые или устаревшие методы шифрования. Следующие строки сделают это для *Apache*:

```
SSLCompression off
SSLProtocol All -SSLv2 -SSLv3
SSLCipherSuite ECDH+AESGCM:EDH+AES
GCM:AES256+EECDH:AES256+EDH
```

Это предотвратит атаки типа *BEAST*, *LOGJAM* и *CRIME*, а также отключит устаревшие SSL v2 и 3. То есть владельцы допотопного *Internet Explorer 6* не смогут просматривать ваш безопасный сайт, однако это, вероятно, не самое большое горе.

Более подробно про настройку — на <http://bit.ly/StrongSSLSecurityOnApache>.

Есть также некоторые заголовки безопасности, принимаемые современными браузерами и способные справиться с некоторыми атаками на ваш сайт — например, вы можете задать

```
Header set X-Frame-Options: sameorigin
Header set X-XSS-Protection: 1;mode=block
Header set X-Content-Type-Options: nosniff
```

Это предотвратит хищение нажатий кнопок [clickjacking] и атаки XSS на ваш сайт, а заодно не позволит браузеру пытаться определить тип MIME скачанных файлов, доверяя вместо этого предоставленной сервером информации.

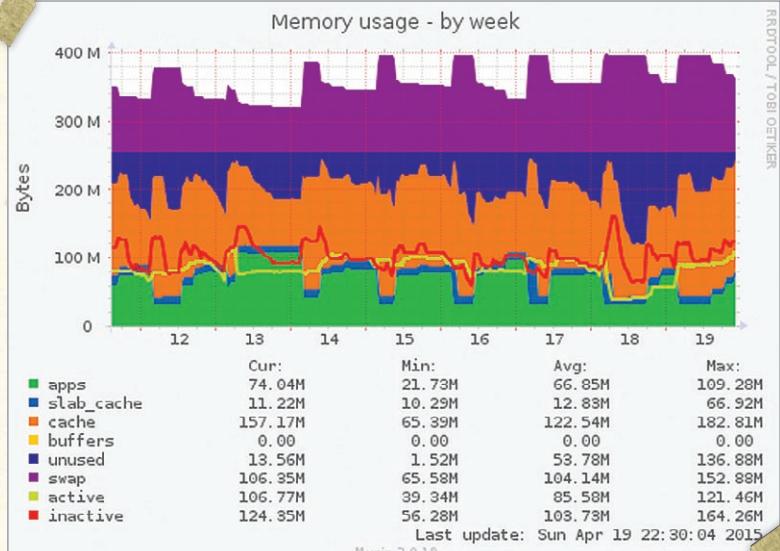
*Fail2Ban* может защитить не только SSH, но и любой сервис: например, вы можете использовать его, чтобы боты перестали лезть на ваш web-сервер. Просто сообщите ему, на какие лог-файлы смотреть, какое количество попыток считать чрезмерным и какой продолжительностью блокировки награждать особо рьяных. *Fail2Ban* внутри использует *iptables*, и трафик с блокированных хостов тихо прекращается, так что их сверхзапросы не будут перегружать ваш компьютер. Загляните во врезку вверху (*Укрепляйте свой web-сервер*), там вы найдете советы по защите web-сервера.

## Тест Kali на вторжение

Упрочение сервера становится совсем другой игрой, когда вы работаете с базами данных и web-приложениями. Такие вещи солидно расширяют область атак, и обеспечение безопасности ваших настроек становится сложнее. Помимо социального инжениринга, другому популярному способу захвата серверов способствует неправильная настройка. Бывает соблазнительно включить в директории политику доступа с широчайшими разрешениями, чтобы работал какой-нибудь экзотический PHP, однако выдача пользователю web-сервера доступа на запись не в те файлы (например, *www-data* в *Debian*) может привести к Очень Плохому Дню для сисадмина.

На *LXFDVD* вы найдете Kali Linux — дистрибутив, разработанный для тестирования на вторжения и оценки безопасности. Все предлагаемые им инструменты можно добавить и в ваш дистрибутив, однако приятно иметь их все в одной посуде. Вы найдете вездесущий *lptar* (и его графический клиент, *zentar*) для просмотра запущенных на машине сервисов, *Aircrack-ng* для взлома беспроводных сетей, *Burp Suite* для испытания на прочность web-приложений, *Metasploit Framework* для тестирования, хм, практически всех известных эксплойтов, и многое другое. Все инструменты разнесены по категориям в меню Kali, и этой страницы не хватит, чтобы достойно о них рассказать, так что обязательно загляните на <http://kalitutorials.net>, там найдется целая серия руководств по Kali.

Может показаться забавным протестировать возможность вторжения на ваш любимый (или, наоборот, ненавистный) сайт, однако на это нужно явно выраженное согласие оператора сервера; если его у вас нет, воздержитесь от подобных развлечений. У вас могут появиться серьезные проблемы, даже если вы не причините никакого ущерба. Свои собственные серверы тестируйте сколько угодно, но если они размещаются на оборудовании, принадлежащем другим людям, владельцы оборудования должны знать о том, что вы намерены делать. Если вы просто хотите поиграть с уязвимыми системами, опций имеется несколько. Одна из них — скачать



Используйте на своих серверах инструменты мониторинга, подобные *Munin*, чтобы выявлять всплески потребления ресурсов — или создавать красочные графики.

очень уязвимое web-приложение *Mutillidae* из проекта OWASP. Вы найдете его на <https://sourceforge.net/projects/mutillidae/>, где есть также ссылки на видео с инструкциями. Другая опция — поиграть с песочницей Acuteneix (<http://testphp.vulnweb.com>), и посмотреть, что случится, если вы перейдете на следующий фальсифицированный URL (пробелы проставлены умышленно):

**Другому популярному способу захвата серверов способствует неправильная настройка.**

```
http://testphp.vulnweb.com/artists.php?artist=-1 UNION SELECT
1,pass,cc FROM users
```

У нас не должно получаться такое вытворять, и именно поэтому так важно вычищать любую введенную пользователем информацию, как только она проникла в ваше приложение. Вам следует использовать такие технологии, как

```
var_dump(filter_var($artist,FILTER_SANITIZE_SPECIAL_CHARS));
```

И вся ситуация в целом намного улучшится.



# Хакерские войны

Будущая правительственная интервенция реально страшнее будущих взломов.

**И**стория программирования должны были дать нам ценный урок: программисты ошибаются (как, впрочем, и вообще люди). Переполнение строки, или динамической области памяти, или буфера существуют давным-давно, и пока во всем этом участвуют люди, не стоит ждать особых изменений. Разумеется, мы можем хвататься за вновь созданные, безопасные для ввода, для памяти и для потока языки типа Go, Rust и Swift, но это лишь породит новые классы ошибок. А плохие парни немедленно найдут новые и оригинальные способы их эксплуатации. Например, в то самое время, когда мы нанесли последние штрихи на эту статью, в OpenSSH была обнаружена новая ошибка (даже две: CVE-2016-0777 и -8).

В начале XX века Никола Тесла написал, что войны будущего будут вести машины и роботы. Если мы признаем за ним здесь некое право на пророчество, то он прав — подумайте о дронах и роботах-бомбардировщиках. Если предположить, что современные тенденции будут продолжаться, то будущее поле битвы переместится в Сеть. Государства разработали «цибер-армии», а на разработку военных «цибер-стратегий» тратятся огромные суммы. Мы видели, что может быть разрушена важнейшая инфраструктура: в декабре 2015 г. оказалось, что аварийное отключение электроэнергии в западной Украине было результатом хакерской атаки

по политическим мотивам. Зачем бомбить город, если можно просто отключить ему воду и электричество?

Злые и добрые хакеры ежедневно обнаруживают все новые и новые возможности взлома. Они могут включать наборы инструментов для атаки на BIOS/UEFI или даже манипулировать памятью вашего ноутбука, пользуясь электромагнитными эффектами между смежными ячейками (как в атаке Rowhammer). Если достаточно мотивированный атакующий (например, государство) особо заинтересован в том, чтобы пресечь онлайн-деятельность

человека, он может использовать атаку нулевого дня. Ноль здесь означает количество дней после обнаружения эксплойта, и если вы уязвимы, вы ничего не сможете

## Шифрование с публичным ключом взъерошило обе стороны Атлантики.

с этим сделать. Однако, при всех их технических возможностях, подобного рода атаки сравнительно редки, как мы уже говорили выше. Несмотря на то, что украинская атака использовала передовое вредоносное ПО, известное как BlackEnergy, главной причиной было целевое фишинг-мошенничество с использованием инфицированного документа Microsoft Word.

## Проблемы шифрования

Тем не менее, правительства все больше интересуются деятельностью исследователей безопасности, а предложенные поправки



## Взломы и статистика — крупнейшие атаки

### Anthem

Американский гигант в области здравоохранения был заражен вредоносным ПО Mivast от группы Black Vine в мае 2014 г., но эта атака обнаружилась только в феврале 2015 г. В результате этой атаки было похищено рекордное количество медицинских карт — 80 миллионов, хотя анализ подсказывает, что эти данные не были первоочередным интересом группы. Согласно Symantec, группа выразила значительный интерес к подрядчикам в области обороноспособности и энергетики, и пыталась найти информацию об этих людях в их медицинских картах. Однако с учетом объема персональных данных, полученных в результате атаки, вполне вероятно, что по крайней мере часть из них всё же окажется на черном рынке и доставит огромные неудобства жертвам.

### OPM

Управление кадрами [Office of Personnel Management] в США ведет учет всех госслужащих. В июне 2015 г. они объявили об утечке данных, которая началась не позднее марта предыдущего года. Согласно оценкам, общее число служащих, подвергнувшихся риску, составило около 21,5 миллиона. Похищенные сведения включали данные о личном составе, зарубежных резидентах и их семьях, а также 5 миллионов наборов отпечатков пальцев.

Затраты на ликвидацию последствий (включая выплаты надлежащих страховок пострадавшим сторонам и отзыв скомпрометированных секретных агентов) предположительно составят миллиарды долларов.

Американские официальные лица сообщили, что атака велась из Китая, однако осторожненько уточнили, что нет никаких свидетельств участия в ней китайского правительства. На самом деле у злоумышленников были подлинные реквизиты доступа, полученные, вероятно, благодаря методам социальной инженерии. Выяснилось, что большая часть данных Управления кадрами хранилась на архаичных мейнфреймовых системах под управлением устаревшего кода COBOL. Это говорит о совершенно другой проблеме в ИТ: когда разбирающиеся во всем этом люди уйдут на пенсию (и в конечном итоге — из жизни), область ИТ заполонят самоувренные милленисты, которые понятия не имеют, как работать с этими системами.

### Hacking Team

Трудно не ухмыльнуться над судьбой компании, продавшей инструменты перехвата данных тоталитарным режимам в таких странах, как Судан, Россия и Саудовская Аравия. Нам-то, конечно, стало об этом известно по единственной причине: кто-то украл у них 400 ГБ документов и затем перехватил

их учетную запись в Твиттере, чтобы поведать об этом миру. Hacking Team позиционируют себя поставщиками средств наблюдения для правоохранительных органов, хотя они многократно повторяли, что не работали со странами, попавшими в черный список. Помимо явного опровержения этого заявления, похищенные данные подтверждают, что Team использовали атаку нулевого дня для заражения целей своим вредоносным ПО RTS. Когда о них сообщили открыто, соответствующие дыры в затронутом ПО (Adobe Flash Player...) были быстро подплатаны, так что уязвимости спалились. Hacking Team продолжает заниматься бизнесом, однако не вполне понятно, в каких масштабах.

### Fiat Chrysler

В принципе, не стоит удивляться, что подключенный к Интернету автомобиль можно хакнуть точно так же, как и компьютер. Однако тот факт, что подобный взлом позволил Чарли Миллеру [Charlie Miller] и Крису Валасеку [Chris Valasek] дистанционно остановить на шоссе джип, отключить его тормоза и запутать передачи, привлек пристальное внимание СМИ и даже обеспечил хакерскому дуэту камео в сериале *CSI: Cyber*. К счастью, никто не пострадал, кроме, пожалуй, Chrysler, которому пришлось отозвать 1,4 миллиона машин, чтобы залатать дыры.

## Equation

к Вассенаарскому соглашению (Wassenaar Arrangement — набор правил, которые подписанты согласились выполнять) усложнят исследователям возможность делиться своими достижениями с другими странами. Создание трудностей в этом исследовании или, что еще хуже, вменение в обязанность сообщать об уязвимостях правительственный органам, а не авторам соответствующих программ, создаст весьма опасный прецедент.

В прошлом году Дэвид Кэмерон [David Cameron] высказал мнение, которое предполагало желание запретить шифрование. Потом его представители заявили, что это не является его позицией, однако предложенный Закон о следственных полномочиях [Investigatory Powers Bill] (в черновом варианте он составляет 300 страниц, на случай, если хотите почитать что-нибудь легонькое) содержит положения, обязывающие интернет-провайдеров по постановлению суда предоставлять данные клиента. Формулировка туманная, однако законопроект ссылается на то, что министр внутренних дел может потребовать «снятия электронной защиты» с провайдеров телекоммуникационных услуг. Когда граждане используют межабонентское шифрование (GPG-почту, например, WhatsApp), только они сами могут расшифровать свою беседу, и ни одна третья сторона не может ничего сделать [Ред.: — По закону вы уже обязаны сообщать все известные ключи государственным органам Великобритании] с любой «электронной защитой».

Так называемое сильное шифрование (раздел математики, именуемый шифрованием с публичным ключом) взъерошило обе стороны Атлантики. Многие политики поддерживают «третейскую» систему с ключами, когда граждане могут свободно общаться конфиденциально, однако у правительства имеется «мастер-ключ» (или иной обходной путь), позволяющий подорвать всю эту конфиденциальность. Технологические фирмы объединили четкое неприятие этой идеи — не потому, что они симпатизируют четырем всадникам кибер-апокалипсиса (террористам, пиратам,

STATUS:	Active	TYPE:	Complex cyberattack platform
DISCOVERY:	2014	TARGETED PLATFORMS:	Windows
FIRST KNOWN SAMPLE:	2002	NUMBER OF TARGETS:	500-1,000



➤ Сайт <https://apt.securelist.com> содержит подробную информацию о высокопоставленном вредоносном ПО.

наркоторговцам и педофилям), а потому, что знают: любой лазейкой с тем же успехом воспользуются и плохие парни.

Представьте, что произойдет, если преступное государство получит ключ от абсолютно всех частных разговоров. Недавняя история показывает, что секреты, даже правительственные, имеют скверную привычку становиться достоянием общественности. Да, преступники используют и будут все больше использовать сильное шифрование, но они также совершают ошибки, которые делают их уязвимыми для вполне традиционных методов расследования. А еще они активно используют поезда [Ред.: — И пушки], однако почему-то никто не заикнулся о том, чтобы поставить и поезда вне закона. [ЛХФ](#)

JuniperOS

Juniper предлагает предприятиям, озабоченным аспектом безопасности, устройства брандмауэра, работающие на ее собственной специализированной ScreenOS. В декабре компания заявила, что в их прошивке обнаружился «несанкционированный код». Это история двух уязвимостей.

Первая была жестко кодированным администраторским паролем (<<< %S(un=' %S') = %U, выбранным, видимо, для мимикрии с дезассемблированным шумом), который, похоже, добавился к концу 2013 г.

Второй было использование одобренного АНБ генератора псевдослучайных чисел Dual\_EC\_DRBG, точнее, его использование с предварительно вычисленными параметрами P и Q. Непонятно, когда (и зачем) ввели этот генератор, однако предполагается, что это было где-то в 2008 г.: после его стандартизации NIST и после того, как сообщество безопасности предупредило, что в нем скрыта потенциальная лазейка.

В теории система предназначалась для того, чтобы смешать результат от Dual\_EC с другим генератором, ANSI X.9.31; к сожалению, кто-то допустил в коде ляп (увеличенный размер случайного числа в DUAL\_EC), и этого не произошло. В конце 2012 г. злоумышленникам каким-то образом удалось изменить значение Q программы на то, которое

позволило им пассивно расшифровать весь трафик VPN. Однако стоит отметить, что изначальная

постоянная Juniper могла бы позволить Juniper сделать то же самое.

# HACKING TEAM RCS

**SUSPECTED GOVERNMENT USERS**



Утверждают, что Hacking Team продала шпионское ПО правительствам из черного списка



# Осторожно, прошивка!

Джонни Бидвелл встретился с ученым и джентльменом Мэттью Гэрреттом, чтобы поговорить о надежности вычислений в пост-Сноуденовскую эру.



Мэттью Гэрретт [Matthew Garret] — главный специалист по защите информации в CoreOS. Как разработчик ядра, он заинтересован в надежности вычислений, и решает проблемы, за которые другие даже не боятся. Кроме того, у него степень PhD в области компьютерной генетики, и он любезно согласился поделиться с нами частичкой своей мудрости в этом интервью.

**Linux Format:** Чем вы занимаетесь в CoreOS?  
**Мэттью Гэрретт:** В основном, инфраструктурной безопасностью: работаю над низкоуровневыми технологиями и средствами обеспечения безопасности как отдельных контейнеров, так и платформы в целом.

**LXF:** Контейнеры сейчас на пике популярности — как по-вашему, уровень безопасности уже тоже соответствует, или появляются лишь новые классы перепрошивок и способов заражения?

**МГ:** Безопасность контейнеров — это интересный вопрос: здесь все зависит от того, как взглянуть. Если вы их используете так же, как виртуальные машины, то контейнеры дают вам меньшую локализацию. Но если вы имеете дело с системой, где ранее все приложения работали в одной среде, то разнеся их по контейнерам, вы повысите их безопасность. Так что контейнеры могут быть как на благо, так и во вред безопасности — то есть «как взглянуть».

**LXF:** Но это все еще неизведанная территория? То есть контейнеры, в форме chroot'ов и всего такого существуют с незапамятных времен, но с такими вещами, как Docker, все это вышло на совершенно новый уровень.

**МГ:** Дополнительные функции ядра, связанные с контейнерами, разрабатывались довольно давно и долго. Нелегко было добиться того, что мы имеем сейчас. В нескольких версиях ядра большинство проблем с безопасностью касались кода пространства имен, служащего основой для контейнеров. Но сейчас все начинает стабилизироваться, вскоре мы сможем говорить о том, что они достаточно надежны.

**LXF:** То есть вы считаете, что мы доросли до контейнеров?

**МГ:** Да, думаю, прошел тот момент, когда это было неким развлечением. Довольно многие сейчас всерьез присматриваются к ним, чтобы использовать их для дела.

**LXF:** Расскажите нам о своем докладе на OSCON.

**МГ:** Я рассказывал о том, что нужно для создания надежного компьютера. Надежного — значит, такого, чтобы его владелец или пользователь были уверены, что он действует в их интересах, а не чьих-то еще. Ведь из сведений, разглашенных Сноуденом, мы узнали, в частности, и о том, что ЦРУ предпринимало различные атаки на встроенное ПО. Например, перехватывая системы на пути к покупателям и меняя прошивку, чтобы иметь возможность управлять ее состоянием. То есть если ЦРУ удавалось получить доступ к системе, они могли и модифицировать встроенные программы, чтобы получить дополнительный контроль. Или же сделать так, чтобы эти программы копировали данные из системы, записывали пароли в энергозависимую память — и тому подобное. Заполучив такую систему, Управление могло выудить из нее все секреты, даже с зашифрованных дисков.

**LXF:** Коммерческие прошивки — в особенности когда беспроводной адаптер не работает прямо из коробки — самый раздражающий фактор для пользователей Linux. Но ведь это и одна из главных проблем безопасности?

**МГ:** Устройства PCI по определению имеют прямой доступ к памяти, чтобы изменять содержимое

ОЗУ. Это было не раз доказано в ходе испытаний, когда разные PCI-устройства вносили в ядро динамические изменения. Противостоять этому можно, указав системному IOMMU (I/O Memory Management Unit), какие именно устройства обладают правом записи и куда. Но IOMMU есть не во всех устройствах, а даже если есть, то дистрибутивы Linux зачастую не позволяют его использовать.

В итоге, при определенном стечении обстоятельств, риск закачать на свое устройство измененную прошивку остается все равно, даже если у вас подлинная ОС и официальные драйверы, и она сможет повернуть что-нибудь в ядре так, что процесс, запущенный обычным пользователем, вдруг получит права root и сможет делать все, что угодно.

**LXF:** Ничего себе. К нашему прекрасному журналу выходит ежемесячное приложение в виде диска, и ежемесячно же поступают жалобы от читателей на UEFI. Мы стараемся объяснить им, что проблема не в самом UEFI и не в Secure Boot, а в том, что производители решили сделать их закрытыми. Расскажите что-нибудь об их положительных сторонах.

**МГ:** В последнее время мы немало поработали над тем, чтобы подтвердить подлинность используемой ОС сделалось проще. UEFI Secure Boot, естественно, была для нас проблемой, ведь изначально риск был в том, что пользователь вообще не сможет контролировать доверительность системы. Была опасность улучшить безопасность ценой ущемления прав пользователей. К примеру, система Secure Boot, разрешающая загружать только подписанные образы Ubuntu и Fedora, возможно, была бы безопаснее, но при этом не давала бы пользователю реализовать его право создавать и выполнять собственный код и работать с ОС по своему выбору.

## ОБ ОПАСНОСТЯХ ИЗМЕНЕННОЙ ПРОШИВКИ

### Процесс, запущенный обычным пользователем, вдруг получит права root.

В итоге мы пришли к тому, что пользователи теперь в состоянии использовать свои собственные ключи и сами решают, что им устанавливать. Хотите — можете взять систему Secure Boot и настроить ее так, чтобы она не давала загружать Windows, а только, скажем, Fedora и Ubuntu, и любой другой ваш любимый дистрибутив. Можно и так; и по-моему, в том и состоит суть свободы: свобода дает решать не только что можно загружать, но и что нельзя. Так мы дошли до уровня ОС. Теперь получить доступ низкого уровня к вашей ОС, тем самым нарушив ее безопасность в целом, стало гораздо труднее. Но мы должны продолжать работать над этим.

**LXF:** А бывали случаи, когда кому-то удавалось обойти UEFI?

# Мэттью Гэрретт

**МГ:** Не знаю, доходили ли до вас какие-то детали из того, что просачивается от Hacker Team, но у них был один способ изменения прошивки системы с помощью их вредоносного ПО. Они применяли технологию довольно низкого уровня, не особо мудреную — им нужен был физический доступ к системе; но если им удавалось заполучить прошивку, то они могли добавлять туда свой код. И при каждой загрузке этой системы он действительно выстраивал жесткий диск с использованием NTFS (код работал только под Windows). Он загружал системный раздел, затем копировал вредоносное ПО в файловую систему Windows, затем размонтировал ее, и при последующей загрузке автоматически это ПО запускал. Ни очистка диска, ни его удаление и замена против заражения не помогали. Сейчас у нас есть инструменты, чтобы сделать это было труднее — в новых системах обновить прошивку можно, если она подписана ключом безопасности. Но это по-прежнему можно обойти, если иметь прямой доступ к флеш-памяти,

то есть или получить доступ к вашей машине и извлечь флеш-чип, или напрямую перепрограммировать прошивку.

**LXF:** Но разве чипы Trusted Platform Module (TPM) не предупреждают вас об этом?

**МГ:** Есть технологии, способные это проверить: с помощью TPM-чипа можно измерить объем встроенных программ при инициализации — при внесении изменений в прошивку, он изменится. Тогда, если у вас есть база правильных значений и вы видите, что они не совпадают, это тревожный знак. Проблема в том, что если вы не доверяете своей прошивке, то откуда вам знать, не заставила ли она вашу ОС прежде выдавать вам верные значения, даже если это не так? Тогда вам понадобятся два устройства, два независимых источника информации.

Я занимался одной технологией использования TPM'ов для создания шифрования с применением двухфакторной аутентификации устройства (2FA).

Таким образом, при каждой загрузке вашего ноутбука TPM будет дешифровать пароль при совпадении значений, и этот пароль, а также время входа в систему, могут составить алгоритм Time-based One Time Password (TOTP). Это будет шестизначное число, которое вы потом можете проверить с телефона; если всё совпадает, значит, ваша прошивка надежна. Если число не совпадает или система не может его воспроизвести, то это повод для беспокойства.

Но даже если вы уверены, что вашу прошивку не меняли, всё еще остается проблема: сам производитель мог намеренно оставить лазейку или просто не заметил уязвимости, которой злоумышленник может воспользоваться, чтобы обойти защиту. И эту проблему нам не решить без исходного кода прошивки — нам нужно досконально знать, что туда включено, нужно изучить этот код. И что еще важнее, нам нужно его воспроизвести и проверить, насколько он соответствует тому, что в прошивке. Будет трудно убедить в этом производителей, но в итоге, это единственный способ утверждать «Я уверен, что этот компьютер работает на меня». Понятно, что это только начало проблем. Допустим, мы убедились в надежности прошивки ОС. Прекрасно, но как насчет прошивки жесткого диска?

**LXF:** Или USB-флешек?

**МГ:** Верно — во многих устройствах есть собственные встраиваемые программы. А если кто-нибудь сможет воспроизвести прошивку вашего жесткого диска, в ходе его работы, и подменить некоторые двоичные файлы — например, запустить SSH, и при первом запуске диск выдаст правильную версию, вы введете свой пароль, и с виду всё будет прекрасно. В ядре тоже можно измерить ее объем через TPM, так что вы можете проверить, не изменил ли кто-то ваш SSH. Но ядро делает это только при первом запуске SSH, либо только зная о внесенных изменениях. Что если при следующем запуске SSH ваш жесткий диск выдаст другую версию, где уже записан ваш пароль? А вы и не знаете — это будет никак не проверить. И вновь, не имея исходного кода прошивки жестких дисков и возможности проверить ее на внесения изменений, мы от этих рисков не уйдем. И это разумно, ведь чем больше открытого кода мы имеем, тем больше мы можем его усовершенствовать и гарантировать его надежность, и тем труднее будет злоумышленникам что-то против него предпринять.

**LXF:** Но разве сами эти системы верификации нельзя взломать — насколько на них вообще можно положиться?

**МГ:** Это тоже одна из проблем — в самих TPM используется проприетарная прошивка, и пока вы можете только уповать на то, что TPM надежны. Для гарантии полной уверенности надо сделать открытыми множество компонентов.

**LXF:** Учитывая, что для некоторых частей кода прошивки такое произойдет довольно-таки не скоро, не делает ли это ваши усилия напрасным трудом?

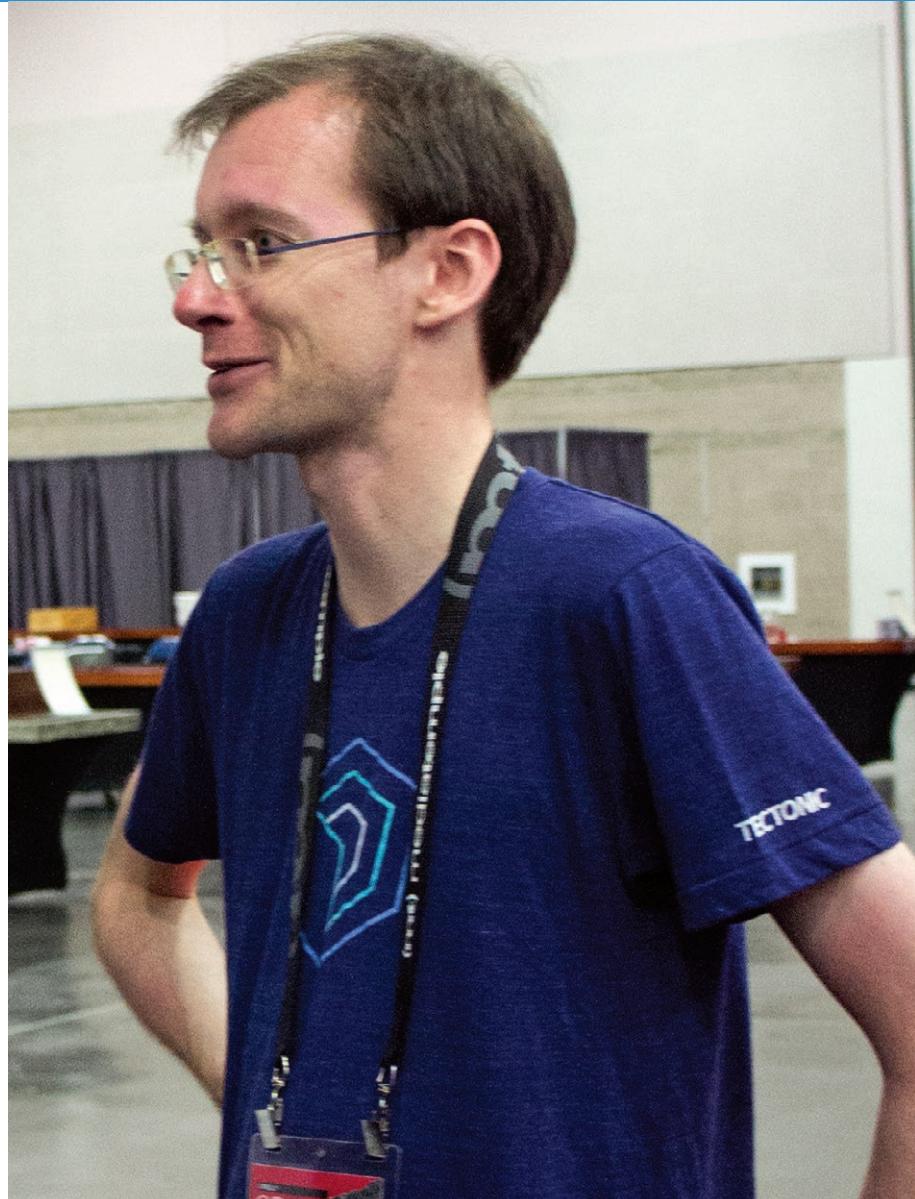


**МГ:** Даже если мы говорим: «Если мы не можем проверить подлинность прошивки жесткого диска, зачем тогда все это нужно?» — ну, хотя бы затем, что теперь им необходимо знать, какой у меня тип жесткого диска; нужно найти уязвимость, которой можно было бы воспользоваться. Осуществить атаку теперь куда труднее, чем раньше. И чем труднее это становится, тем больше вероятность, что взломщики станут прибегать вместо этого к другим старомодным способам. Разумеется, если АНБ или Центр правительственной связи захотят узнать, что вы делаете со своим компьютером, то они наверняка смогут это сделать. Но на каком-то этапе, каждый раз, когда они пользуются этими уязвимостями, они рискуют быть обнаруженными. В самом крайнем случае, даже если АНБ сможет заставить Intel внедрить уязвимость в их процессоры, каждый раз, как они будут ею пользоваться — а им придется это делать, поскольку она прописана в коде — есть шанс, что кто-нибудь догадается, что дело именно в процессоре. Найдется кто-нибудь особо дотошный, начнет записывать команды, отправляемые процессору, и заметит, что тот выдает неверный результат. Или что случайные числа, так или иначе, детерминированы. Если это вскроется и будет доказано, продажи Intel за рубежом упадут в разы, принеся правительству США большие проблемы.

**LXF:** Зато в AMD, при таком раскладе, порадуются.

**МГ:** Да, в AMD порадуются, если доживут до этого момента, но это был бы сильнейший удар по имиджу американской индустрии технологий. В реальности, это все равно, что представлять непосредственную угрозу существованию государства. В таком масштабе, это должно волновать не меньше, чем какой-нибудь главарь террористов. Чем активнее мы будем с этим бороться, тем защищеннее люди будут себя чувствовать, даже против надзора со стороны государства, так что наше дело правое.

**LXF:** Очевидно, что открытого ПО сегодня море, но с аппаратными средствами — даже



со встраиваемыми программами, на них работающими — всё совсем иначе?

**МГ:** Да. В этом плане все далеко не так, как бы мне хотелось. Программы трудно писать, но в готовом виде их легко скопировать или воспроизвести. Сам по себе процесс создания ПО не является секретом. А если вы захотите создавать процессоры, способные конкурировать с Intel'овскими, то даже если вы сможете такой спроектировать, для производства вам все равно не обойтись без привлечения ресурсов других компаний. И как убедиться в том, что они подлинные?

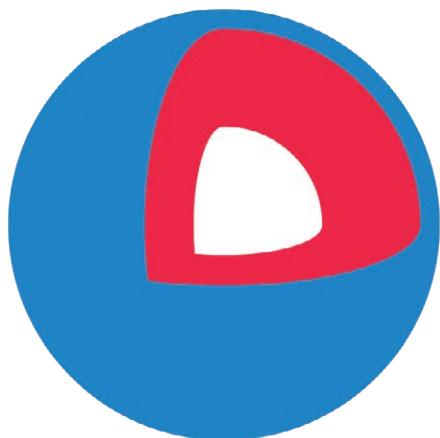
Здесь есть масса уровней, где вероятно вмешательство, и масса способов для его осуществления. Построить достаточно открытую схему производства и сказать «Я на 100 % уверен, что это то самое оборудование, которое хотел создать», очень и очень трудно.

**LXF:** Мы интересовались ноутбуками вроде Novena и Purina, которые стремятся быть настолько открытыми, насколько можно рассчитывать

**О БОРЬБЕ С ЗАКРЫТЫМИ ПРОШИВКАМИ  
ДЛЯ ПОЛНОЙ УВЕРЕННОСТИ  
НАДО СДЕЛАТЬ ОТКРЫТЫМИ  
МНОЖЕСТВО КОМПОНЕНТОВ.**

в реальности. Но, похоже, и там есть свои кирпичные стены...

**МГ:** Я общался с людьми из обоих проектов, с этим до сих пор есть проблемы. В Novena системная прошивка открыта и полностью доступна, кроме системы-на-чипе (SoC), на котором она работает. Сегодня в SoC ARM есть ядра ARM, на которых вы вполне можете запустить свою ОС, но есть и другие, гораздо более мелкие, отвечающие за управление питанием, поддержку вспомогательного оборудования — к примеру, у SATA-контроллера может быть свое ядро. И во многих из них будет закрытая прошивка, встроенная в сам чип — код, который не предполагается менять в течение всей жизни устройства. Даже если мы сосредоточимся только на системной прошивке, останутся еще и прочие компоненты, которым мы вынуждены довериться. И с этим ничего не поделаешь. **LXF**



# Core OS



# МУЛЬТИЗАГРУЗКА С GRUB

Широкий выбор позволяет покоризничать.  
**Нейл Ботвик** показывает, как заполучить  
на компьютер несколько дистрибутивов.

**Д**истрибутивов Linux полным-полно, и каждый имеет свои сильные и слабые стороны. Когда вы хотите попробовать другой дистрибутив, обычно вы можете получить live-версию или установить его на виртуальной машине для беглого теста. Оба варианта простые и быстрые, но это не то же самое, что на прямую установленный дистрибутив.

Но, возможно, вы не намерены отказываться от своего привычного дистрибутива; или, может, делите с кем-то компьютер, и вы предпочитаете Mint, а этот кто-то — Fedora; было бы отлично иметь установленными оба. Итак, что же делать? Обычно термин «двойная загрузка» относится к ситуации, когда у вас на одном компьютере установлены дистрибутив Linux и Windows, и вы выбираете

между ними во время загрузки; но то же самое возможно и с двумя дистрибутивами Linux. На нескольких следующих страницах мы рассмотрим, как настроить компьютер, чтобы загружаться с выбором нескольких ОС, одной из которых может быть Windows, и чтобы вам было доступно больше одного дистрибутива Linux. Вы даже можете пойти дальше и включить одну или несколько ОС BSD. Мы также рассмотрим, как распределить ваши документы и фотографии между разными используемыми вами дистрибутивами.

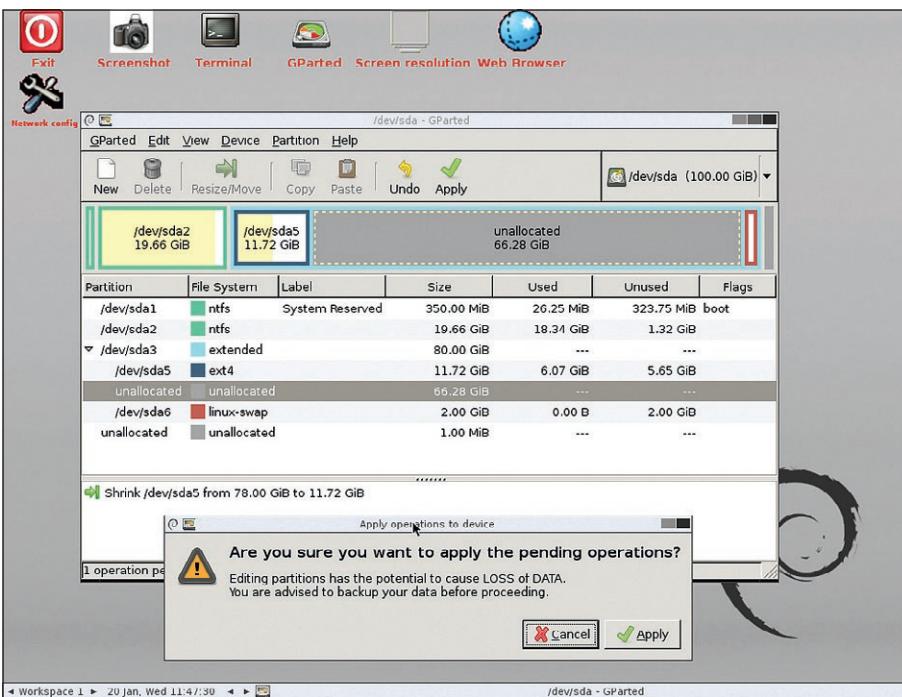
**Мультизагрузка Linux стала проще благодаря внедрению Grub 2.**

## Grub 2 против старого Grub

Мультизагрузка Linux упростилась благодаря почти повсеместному внедрению загрузчика Grub 2. Доступно две основных версии Grub. Старую версию, так и не достигнувшую номера 1.0, часто называют «унаследованным [legacy] Grub», а подавляющее большинство дистрибутивов сейчас использует более новый Grub 2. Grub 2 очень сильно отличается от старой программы и получил репутацию сложного. На самом деле, его модульный принцип означает, что он может справиться с куда большим

числом ситуаций при загрузке и чаще всего способен автоматически настроиться самостоятельно. Далее мы будем рассматривать только Grub 2, называя его просто Grub.

У Grub есть три основные части. Код начальной загрузки обычно загружается



➤ **GParted** — простейший способ управлять разделами, создавая пространство для другого дистрибутива

в Главную Загрузочную Запись (MBR) диска. Это небольшое пространство, 446 байт, то есть этот код минимален, и его хватает лишь для загрузки второй части, находящейся в вашей директории **boot** или разделе в директории под названием **grub**. Здесь вы обнаружите различные модули файловой системы вместе с темами и шрифтами, используемыми в случае, если ваш дистрибутив индивидуализировал вид загрузочного меню. Вы также найдете наиболее важный для данной статьи файл: **grub.cfg**. Этот файл содержит описания меню, опций, которые появляются в загрузочном меню, и того, что происходит при выборе каждой из них.

Первый шаг — установить несколько ОС. Если вы хотите включить Windows в ваш список ОС, она должна быть установлена первой — обычно это не является проблемой, поскольку весьма вероятно, что на компьютере она уже была предустановлена. Установщики Linux хорошо определяют существующую установку Windows и работают с ней. Затем установите предпочитаемый вами дистрибутив обычным способом. Это предоставит вам стандартную двойную загрузку; если она у вас уже имеется, можете пропустить предыдущий параграф и сразу перейти к интересной части добавления дополнительных дистрибутивов Linux к вашему раскладу.

## Добавление дистрибутива

Установщики дистрибутивов переразбивают жесткий диск с разной степенью успеха, так что лучше будет подготовить свой жесткий диск заранее. Лучший инструмент для этой цели — **GParted**, причем — какое совпадение! — его свежая версия попала на **LXFDVD** данного номера. Загрузитесь в **GParted Live** и измените объем вашего раздела **root** на подходящий. **GParted** подскажет вам, как далеко вы можете зайти, но по возможности сделайте его как минимум на 50% больше, чем нужно сейчас. Не создавайте раздел в высвобожденном месте, оставьте его нераспределенным, затем установите второй дистрибутив обычным способом, велев ему использовать нераспределенное место на диске. Установка проходит в обычном режиме, но с одним исключением — вам не надо устанавливать **Grub** на MBR. В большинстве установщиков есть возможность выбрать расположение **Grub**, иногда упрятанная за кнопкой **Advanced**. Если это не так, попозже мы научим вас перемещать его. Выберите либо его установку в раздел **root**, либо вообще не устанавливайте его. Это затронет только первую часть кода **Grub**, файлы в **boot** и других местах все равно будут установлены. Если вам предлагается выбрать раздел **swap**, выберите его из другого дистрибутива, и они смогут использовать его вдвое.

## Кто в доме хозяин

Первый установленный вами дистрибутив должен считаться вашим главным дистрибутивом; это он будет управлять загрузкой, по крайней мере, на данный момент. Именно поэтому главный дистрибутив категорически нельзя удалять, иначе ваш компьютер может сделаться незагружаемым —

по крайней мере, пока вы не загрузитесь с восстановительного диска, чтобы исправить дело. При удалении остальных дистрибутивов разве что станут лишними их пункты загрузочного меню, ведущие в никуда (излечивается новым запуском **update-grub**).

Когда установщик завершит свою работу, перезагрузите компьютер, и вы увидите загрузочное меню первого дистрибутива, где отобразится Windows, если она установлена, однако нового дистрибутива не будет и следа. Причина в том, что вы не трогали настройки **Grub**. Одной из приятных особенностей **Grub 2** является его способность генерировать свои собственные файлы конфигурации, так что откройте терминал и запустите

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

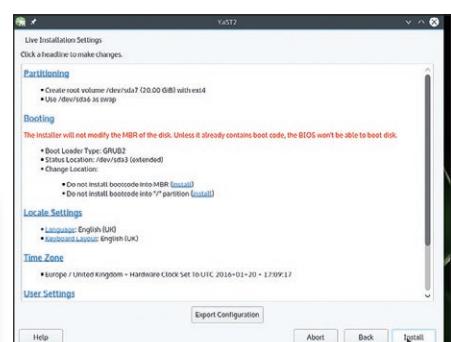
Ваш жесткий диск просканируется на предмет наличия ОС, и для каждой найденной создастся пункт меню. Теперь после перезагрузки вы должны увидеть оба дистрибутива и, может быть, Windows в вашем загрузочном меню. В Ubuntu и его производных есть команда **update-grub** — это односторонний скрипт оболочки, запускающий **grub-mkconfig**, как и приведенная выше команда; выбирайте любой из этих вариантов. Одним из преимуществ отказа от использования скрипта является возможность предпросмотра меню по **\$ sudo grub-mkconfig less** — вы увидите, что будет выбрано и написано в меню. Вы должны понимать «грубояз», чтобы разобраться в этом.

## Перемещение Grub

Если ваша новая установка не предлагает вам переместить **Grub**, вы, возможно, получите загрузочное меню, где уже будет все необходимое, поскольку **grub-mkconfig** запускается как часть процесса. И почему бы не дать этому возможность случиться каждый раз? Проблему создают обновления: когда менеджер пакетов дистрибутива устанавливает обновление в ядро Linux, он снова включает версию **Grub** этого дистрибутива, так что вы найдете меню переключающимся с одного дистрибутива на другой. Чтобы переместить **Grub** в раздел дистрибутива, вначале загрузите дистрибутив, в котором вы хотите управлять **Grub**, и убедитесь, что он работает, с помощью такой команды терминала (при условии, что вы используете диск на **/dev/sda**):

```
$ grub-install /dev/sda
```

Затем загрузитесь в другой дистрибутив и определите раздел, хранящий вашу файловую систему **root**, командой **\$ findmnt -o SOURCE**, и велите **Grub** хранить загрузчик там с помощью **\$ grub-install --force /dev/sdAN**, где **sdaN** — раздел диска **sda**, возвращенный **findmnt**. **--force** вставлен потому, что



➤ Некоторые дистрибутивы позволяют держать **Grub** подальше от MBR, когда вы их устанавливаете, но могут предупредить вас, что это не лучшая идея!

# Мультизагрузка

установка *Grub* в раздел в наше время считается далекой от идеала, но по сути нам просто надо держать *Grub* в сторонке: тогда обновление ядра для этого дистрибутива не испортит загрузочное меню. *Grub* при этом вообще не затронется, так что вам понадобится загрузиться в ваш основной дистрибутив и снова запустить *grub-mkconfig* или *update-grub*, чтобы учсть изменения.

## Настройка *Grub*

Одно из убойных преимуществ *Grub* — его способность генерировать собственное меню, основываясь на содержимом вашего жесткого диска; но процесс создания этих меню можно и настроить, используя скрипты в */etc/grub.d* и настройки в */etc/default/grub*. Этот файл содержит ряд определений переменных, которые можно изменить с целью переделки меню. Например, если вы ничего не выбрали, сам *Grub* обычно загружает первую опцию — найдите строчку, содержащую *GRUB\_DEFAULT=0*, и измените 0 на номер, который хотите видеть по умолчанию. (*Grub* считает от нуля, так что стандартная настройка загружает первый

элемент). Вы также можете изменить время ожидания, в строке *GRUB\_TIMEOUT*, и настройки ядра по умолчанию, в *GRUB\_LINUX\_DEFAULT*. Файл содержит комментарии-пояснения для опций; либо почитайте info-страницу *Grub*, где все опции перечислены со всеми подробностями.

Файлы */etc/grub.d* — это скрипты оболочки, запускаемые по *grub-mkconfig*. При желании перенастроить свое загрузочное меню вы можете добавить свои собственные скрипты: все, что им надо делать — это выводить действующие пункты меню. Скрипты в */etc/grub.d* запускаются по порядку, именно поэтому их имена начинаются с чисел. **00\_header** пишет стандартные настройки в верхней части файла меню, а **10\_linux** создает пункты меню для запущенного дистрибутива, тогда как **30\_os-prober** сканирует ваш жесткий диск на наличие

других ОС, Linux или других, и добавляет их в меню. Что и дает способ содерхать в вашем меню все ваши дистрибутивы.

## Цепочечная загрузка

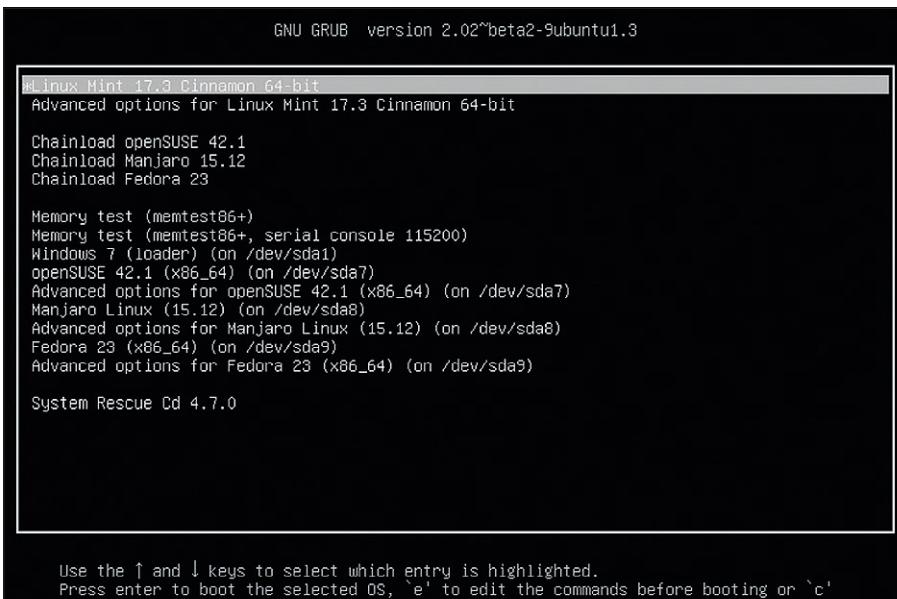
Есть еще один способ управления множественной загрузкой, под названием «цепочечная (последовательная) загрузка». Это тот способ, которым *Grub* загружает Windows, поскольку сам он Windows загрузить не может. Вместо этого он передает управление загрузчику Windows, как будто бы он был загружен напрямую BIOS. Мы можем воспользоваться этим, чтобы выбирать дистрибутивы из начального меню *Grub*, и у каждого дистрибутива будет свое собственное загрузочное меню.

Это означает, что вам требуется способ создавать свои пункты меню. Их нельзя просто добавить в файл *grub.cfg*, поскольку он будет перезаписан при следующем запуске *grub-mkconfig*; но в */etc/grub.d* имеется файл с именем **40\_custom**, пригодный для добавления своих личных пунктов меню. Сделайте

его копию, дав ей осмысленное имя, и, возможно, измените номер, чтобы он включился в меню раньше. Отредактируйте этот файл и припишите действующие пункты меню к низу файла. Не трогайте существующее содержимое — хотя его стоит прочесть! Желая загрузить меню для openSUSE, установленном на */dev/sda7*, при условии, что вы установили *Grub* в *sda7* или переместили его, как показывалось выше, добавьте в файл вот что:

```
menuentry "Load openSUSE boot menu" {  
    set root=(hd0,7)  
    chainloader +1  
}
```

Помните, что *Grub* нумерует диски с нуля, а разделы с единицами, так что *sda7* становится *hd0,8*. Это даст вам собственное загрузочное меню для каждого дистрибутива, так что не надо будет перезагружаться в главный дистрибутив для обновления загрузочного меню; но тогда для загрузки любого дистрибутива, кроме главного, вам придется проходить через два меню. Если вы используете этот способ, то увидите, что у вас по-прежнему есть пункты меню, сгенерированные *grub-mkconfig*. При отсутствии у вас Windows можете избавиться



Ага! Четыре дистрибутива, Windows и пункт загрузки с ISO-образом спасательного CD — все в одном меню, и можно войти в личные загрузочные меню дистрибутивов.

## Системы восстановления

Одной из замечательных функций *Grub 2* является возможность загрузки прямо с ISO-образа. Благодаря этому некоторые журналы могут выпускать отличные мультизагрузочные диски, а вы можете получить спасательный диск или live CD, всегда готовый к загрузке. Мало того, что это быстрее, чем загрузка с настоящего CD/DVD (или даже флешки) — заодно экономится время, которое вы потратили бы на рытье в столе в поиске нужного CD.

Для этого требуется, чтобы дистрибутив поддерживал загрузку с ISO. Большинство ее поддерживают, хотя синтаксис может варьироваться. Всё, что

вам надо сделать — это создать копию **40\_custom** и добавить соответствующее определение меню. Вот пример спасательного диска (я всегда держу его ISO в **boot**):

```
set root=(hd0,1)  
menuentry "System Rescue CD 4.7.0" {  
    loopback loop /systemrescuecd-x86-4.7.0.iso  
    linux (loop)/isolinux/vmlinuz.efi file=/cdrom/preseed/  
    ubuntu.seed boot=casper isoscan/filename=$isofile  
    quiet splash ---  
    initrd (loop)/casper/initrd.lz  
}
```

А вот пример образа live CD для Ubuntu

```
set root=(hd0,1)  
isofile=/Ubuntu/ubuntu-15.10-desktop-amd64.iso  
loopback loop $isofile  
menuentry "Ubuntu 15.10" {  
    linux (loop)/casper/vmlinuz.efi file=/cdrom/preseed/  
    ubuntu.seed boot=casper isoscan/filename=$isofile  
    quiet splash ---  
    initrd (loop)/casper/initrd.lz  
}
```

Обратите внимание на использование переменной *isofile*: работают оба метода, но этот проще в обслуживании.

от этих пунктов с помощью следующей настройки в `/etc/default/grub`:

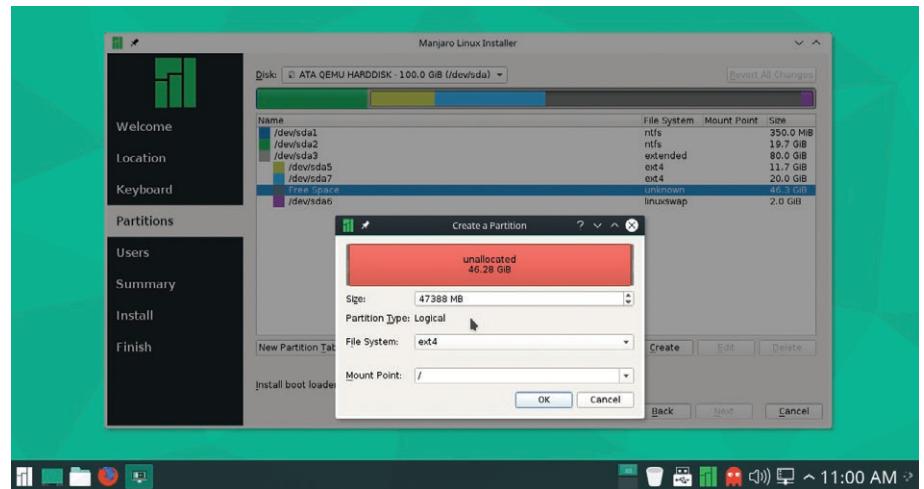
```
GRUB_DISABLE_OS_PROBER=true
```

Поскольку функция `os-prober` добавляет еще и Windows, вы не можете сделать этого, если собираетесь загружать Windows, так что либо переименуйте файл, содержащий пункты цепочечной загрузки, чтобы они появлялись раньше других, нечто вроде `20_chainload`, либо скопируйте пункты Windows из уже существующего `grub.cfg` в свой файл цепочечной загрузки и затем отключите `os-prober`.

## Дележка пространства

Итак, у нас есть несколько мирно сосуществующих дистрибутивов; а как насчет наших данных? Нужна ли нам отдельная директория `home` для каждого дистрибутива? Короткий ответ на этот вопрос — да. Можно, конечно, иметь отдельную файловую систему для `home` и использовать одно и то же имя пользователя и директорию `home`, но это чревато конфликтами. В директории `home` программы хранят свои файлы настройки, и если два ваших дистрибутива имеют разные версии одной и той же программы, у вас начнутся проблемы. Большинство программ радостно прочитают настройки от старой версии и обновят их, но когда вы переключитесь обратно на дистрибутив с более старой версией, программа сломается.

Одно из решений — иметь отдельную файловую систему для ваших файлов данных, т.е. того, что занимает место и является файлами, к которым вы хотите иметь доступ во всех дистрибутивах. Это может быть полностью отдельная файловая система, но может быть и ваша директория `home` в вашем основном дистрибутиве — просто помните, что в этом случае придется немало возиться с файлами, прежде чем удалять дистрибутив, вышедший у вас из милости. Для этого



Может оказаться, что ваш установщик позволяет использовать не все доступное место, избавляя вас от дальнейшей перенастройки объемов разделов в *GParted*.

надо вернуться в *GParted* и изменить размер ваших разделов, создав пространство для большого раздела с данными. Затем отредактируйте `/etc/fstab` в каждом дистрибутиве, чтобы монтировать эту файловую систему во время загрузки. Кстати, стоит добавить пункты `fstab`, чтобы смонтировать другие дистрибутивы в каждом из них, скажем, в `/mnt/distroname` — это упростит подобные штуки, поскольку вы сможете проделать всю работу в одном дистрибутиве. Это также упрощает доступ к файлам из других дистрибутивов. Так что смонтируйте эту новую файловую систему, например, в `/mnt/common` и создайте там по директории на каждого пользователя. Потом в других ваших дистрибутивах можно будет создать сюда символические ссылки, например:

```
$ ln -s /mnt/common/user/Documents /home/user/Documents  
$ ln -s /mnt/common/user/Music /home/user/Music
```

и т.д. Теперь, когда вы сохраните файл в `Documents`, он переместится в `/mnt/common/Documents` и будет доступен всем дистрибутивам. Учтите: здесь предполагается, что вы единственный пользователь компьютера!

## Кто чем владеет?

Теперь вам надо решить щекотливый вопрос о собственности и правах доступа к файлам. Первым делом убедитесь, что у директорий в `/mnt/common` правильные владельцы, используя

```
$ sudo chown -R username: /mnt/common/User
```

Вы можете ожидать, что это сработает на всех дистрибутивах, если в каждом из них создать пользователей с такими же именами; а вот и нет. Дело в том, что файловые системы Linux не волнует имя пользователя — их волнуют числовые ID пользователей (UID). Большинство дистрибутивов дают первому пользователю UID 1000, но парочка все еще начинает с 500, так что проверьте свой UID в каждом дистрибутиве командой `id` — просто запустите ее в терминале без аргументов. Если все UID совпадают, то и отлично; иначе понадобится изменить все несоответствующие UID, редактируя `/etc/passwd`. Никогда не редактируйте этот файл напрямую: ошибка может привести к невозможности входа в систему. Используйте для этого команду `$ sudo vipw`.

Найдите строку для своего пользователя, которая будет выглядеть примерно так:

```
user:x:500:100::/home/user:/bin/bash
```

Первое число — UID; измените его, приведя в соответствие с другими дистрибутивами, и сохраните файл. Затем надо будет сменить владельца всех файлов, принадлежащих старому UID, на нового. Поскольку в вашей директории `home` всё должно принадлежать вам, смело идите в лобовую атаку и смените вообще всех пользователей:

```
$ cd  
$ sudo chown -R username: .
```

Теперь вы можете переключаться между дистрибутивами при каждой перезагрузке, причем каждый дистрибутив будет работать в своем формате, на полной скорости и с полным доступом ко всем вашим данным.

```
*fstab (/etc) - gedit
File Edit View Search Tools Documents Help
Open Save Undo Cut Copy Paste Find Replace Selection
*fstab x
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda5 during installation
UUID=5743a... / ext4 errors=remount-ro 0 1
# swap was on /dev/sda6 during installation
UUID=be7af... none swap sw 0 0
/dev/fd0 /media/floppy0 auto rw,user,noauto,exec,utf8 0 0
/dev/sda6 /mnt/common ext4 noatime 0 0
/dev/sda7 /mnt/opensuse ext4 noatime 0 0
/dev/sda8 /mnt/manjaro ext4 noatime 0 0
/dev/sda9 /mnt/fedora ext4 noatime 0 0
/dev/sda1 /mnt/windows ntfs defaults 0 0

Plain Text Tab Width: 8 Ln 18, Col 49 INS
*fstab (/etc) - gedit
LX Terminal
```

Упростите управление: добавьте в `/etc/fstab` пункты монтирования разделов `root` ваших дистрибутивов.



## Джолион Браун

В свободное от консультаций по Linux/DevOps время Джолион обуздывает стартап. Его самая большая амбиция — найти причину пользоваться *Emacs*.

## Данные как искусство, и наоборот

**П**о части изящных искусств и литературы, лог-файлы моего web-сервера — не то место, куда я в первую очередь отправлюсь за вдохновением. Но в самом конце прошлого года на жестких дисках моих (и миллионов других) серверов появился загадочный текст такого содержания:

«УДАЛИТЕ свои лог-файлы. Удалите свои системы. Сотрите все начисто. Выйдите на тропинку меж цветущих вишневых деревьев, и пусть ваши материнские платы ощутят почву. Пусть вода заструится по корпусу. Вы знаете, что хотите чего-то большего, и я говорю вам: мы любим вас. У нас есть нечто большее. Мы знаем, что вы там, в пустой серверной комнате, полискиваете и мигаете огоньками и никогда не спите. Мы знаем, что вы готовы и ждете. Идите к нам. <3>

Этим удивительным фрагментом поэзии мы обязаны группе, называющей себя masspoem4u, посетившей ежегодный Конгресс по хаосу в коммуникациях (Chaos Communication Congress) в Гамбурге, Германия. Они использовали утилиту *masscan* (она доступна в Github, но обращайтесь с ней осторожно — <http://bit.ly/Masscan>) для распространения своего маленького образчика творчества по всему миру.

Эта идея (столь непохожая на обычных роботов и автоматизированных попыток взлома, которые я обычно вижу в лог-файлах) мне по-настоящему понравилась и напомнила о том, что неплохо было бы попасть на одну такую конференцию (C3, DEFCON, HOPE) в ближайшие год-два (увы, C3 проводится сразу после Рождества, и семья, естественно, в восторг не придет).

Интересно, много ли других сообщений в лог-файлах или в исходном коде годами остаются незамеченными и жаждут увидеть свет? И сколько еще мелких сервисов спят, ожидая, пока кто-то их обнаружит (советую прямо сейчас набрать команду *telnet towel.blinkenlights.nl?*)?

[jolyon.brown@gmail.com](mailto:jolyon.brown@gmail.com)

# По советам м-ра Брауна

Эзотерическое системное администрирование из таинственных закоулков серверной.



## Горести SSH для админов

В клиентах SSH обнаружен код «роуминга», который отправляет закрытые ключи на вредоносные серверы.

**В** январе 2016 года в коде OpenSSH была выявлена довольно туманная, но неприятная ошибка — из серверной части ПО ее давно удалили, но она осталась в клиентской. Код, обнаруженный компанией Qualys, в некоторых случаях позволял SSH-серверу злоумышленнику похищать закрытые ключи клиента. Это было возможно благодаря недокументированной возможности SSH — «роумингу [Roaming]», которая должна была упрощать возобновление временно прерванных сеансов, если обе стороны это поддерживают. Эта ошибка затронула все версии клиента OpenSSH, начиная с 5.4, но после обнаружения для нее быстро выпустили исправление. При отключении недокументированного параметра *UseRoaming* по в файле конфигурации SSH на клиенте (*/etc/ssh/ssh\_config* или *~/.ssh/config*) вредоносный код отключается (Подробности см. в CVE 2016-0777 и 2016-0778). Также неплохо разделить ключи и использовать различные пары для различных хостов (или, на худой

конец, для различных групп хостов). Также стоит использовать *ssh-agent*, установить параметр *IdentityFile* для каждого целевого хоста и добавить директивный параметр *IdentitiesOnly yes* (два последних параметра можно найти в файле *.ssh/config*).

Другая подобная история коснулась устройств для промышленной безопасности от компании Fortinet — в старых версиях прошивок для этих устройств были обнаружены жестко закодированные логины SSH. Системы, не получившие исправлений, могут стать жертвой уязвимости, обнародованной неизвестными лицами, которых изо всех сил вышаривают различные автоматизированные скрипты. Эти логины, похоже, остались от какого-то механизма управления аутентификацией и были удалены в более новых версиях прошивок. Эта проблема доставила неприятности Juniper, который претерпел неавторизованные изменения в коде, открывшие несколько лазеек: одна из них — это жестко закодированный пароль, другая — более тонкая проблема, которая включает использование Dual EC DRBG (криптографически стойкого генератора случайных чисел) АНБ. Исследователи обвинили этот алгоритм в том, что он разрешал достаточно умному злоумышленнику прослушивать зашифрованный сетевой трафик. **LXF**, как всегда, советует оперативно устанавливать исправления и пользоваться принципом наименьших привилегий, когда дело касается открывания портов для сетевого трафика.



➤ OpenSSH — одно из основных средств, используемых сисадминами, но при этом не стоит пренебрегать регулярным обновлением (а зато стоит подкинуть проекту деньжат).

# Rancher: Ваш личный сервис контейнеров

Следует признать: применение инфраструктур с контейнерами скоро будет повсеместным. Ваше бремя облегчит Rancher.

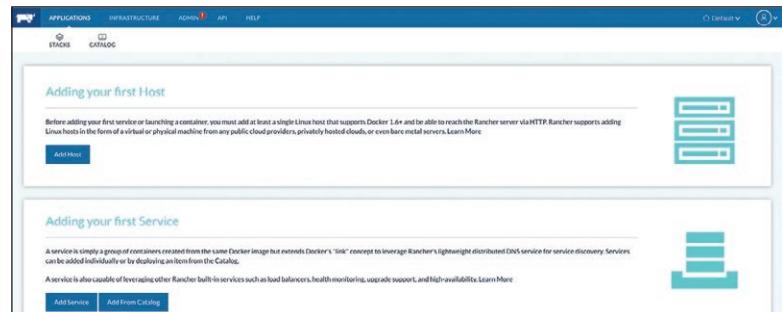
**Я** периодически писал о контейнерах (в CoreOS с Kubernetes) на священных страницах этого журнала, одновременно следя за тем, не утомили ли контейнеры вас, дорогие читатели (вас обоих). Но я не буду извиняться за то, что в этом году контейнеров на этих страницах станет еще больше. В этой рубрике я в основном пишу про то, с чем я сам сталкиваюсь в своей работе консультанта последние месяцев шесть — а стало быть, это популярно на рынке, и информация о востребованных знаниях должна быть полезной. На данный момент работа с контейнерами занимает не менее 50% моих оплачиваемых рабочих часов в месяц.

Эта сфера интересна большим числом происходящих в ней изменений, и большой влиятельностью в ней открытого ПО. В начале моей карьеры сложно было бы представить себе ту свободу и возможности, которые сейчас есть у администраторов, когда дело касается создания инфраструктуры. В этой сфере есть большая конкуренция, и по-моему, это хорошо, особенно потому, что большая ее часть происходит открыто и в результате исходный код становится доступен всем нам. Будет интересно взглянуть, сможет ли какой-то поставщик ПО с закрытым исходным кодом вообще войти на этот рынок. Одна из компаний и групп в этой сфере — Rancher, и их программному обеспечению я посвящу несколько следующих статей.

## На ранчо

*Rancher* — программная платформа с открытым кодом, которая реализует специализированную инфраструктуру для запуска контейнеров в рабочей среде. Контейнеры прекрасно подходят для быстро и воспроизводимого развертывания служб, но не защищены от тех же ограничений планирования, которые свойственны более традиционным моделям. Конечно, выбросить «чистые» экземпляры *Docker* из облака, не заботясь о том, где они окажутся, весьма привлекательно, но над любой инфраструктурой должен быть контроль, хотя бы для того, чтобы сохранить рассудок бедолаг... в смысле, системного администратора, который за ней присматривает.

Как и ряд аналогов, *Rancher* целился на этот рынок, повышая ценность контейнеров *Docker*. Проект состоит из двух элементов/продуктов: *Rancher*, который отвечает за оркестрацию компонентов (это главная часть предложения), и *RancherOS*, максимально урезанного дистрибутива Linux, чисто для запуска контейнеров



*Docker* (отсюда очень большое сходство, например, с CoreOS). Однако *Rancher* может и сам работать на любой целевой платформой с Linux (если она поддерживает *Docker*). Эти два элемента существуют отдельно друг от друга, но рекомендуется, конечно, запускать *RancherOS* как часть развертывания *Rancher*. Сам *Rancher* можно запускать в *RancherOS*.

*RancherOS* «весит» около 30 МБ. Этот крошечный размер объясняется тем, что из системы удалили всё, кроме основ, но оставили возможность запуска *Docker*. Мне нравится, когда установка сервера имеет минимум возможностей для атаки любыми потенциальными злоумышленниками, и этот вариант я нахожу довольно привлекательным. На самом деле, стоит выполнить команду `ls -l` для каталога `/usr/bin`, и мы увидим основу *RancherOS* — *Busybox*. Все в *RancherOS* представляет собой контейнеры *Docker*, включая все известные процессы, которые вы ожидаете найти в Linux. При этом в системе нет *init* или *Systemd* — их заменили на то, что в *Rancher* называется «системным Docker», демоном *docker*, запущенным с идентификатором процесса 1. Он, в свою очередь, запускает специальный контейнер под названием *user-docker*, с которым мы будем сотрудничать при запуске собственных контейнеров. Он также создает хороший уровень изоляции и позволяет избежать влияния на системные сервисы слишком активного пользователя.

*Rancher* — самая весомая и самая яркая часть пакета, которая делает эту систему достойной внимания. Обещано, что *Rancher* может управлять хостами контейнеров и следить за ними, где бы они ни находились. *Rancher* использует приватную сеть, запускаемую

➤ Начальное окно *Rancher* — обратите внимание на то, что при первом входе пользователи должны настроить защиту системы.

## Как настроить RancherOS?

*RancherOS* использует для настройки *cloud-config*, утилиту, которая является довольно распространенным стандартом в Linux и обычно распознается при запуске поддерживающим дистрибутивом.

Ее файлы находятся в каталоге `/var/lib/rancher/conf/`. Но для изменения этой конфигурации рекомендуется запустить команду `ros`: например, `$ sudo ros config export` выведет текущую конфигурацию в командную строку. Если добавить к этой команде флаг `--full`, выведется еще больше информации (внешние и внутренние настройки), например,

```
настройки system-docker и т.п. Формат конфигурации cloud-config — YAML, и поэтому он довольно легко воспринимается человеком. Все параметры имеют ключ 'rancher', который ros может настраивать с помощью подкоманд set и get. Например, просмотреть DNS для хоста можно командой
```

```
$ sudo ros config get rancher.network.dns.nameservers
A назначить другой — командой
$ sudo ros config set rancher.network.dns.nameservers [8.8.4.4, 8.8.8.8]
```

Команда `ros` также может использоваться для обновлений *RancherOS* с помощью команд `$ sudo ros os list`, которая выводит доступные версии, и `$ sudo ros os upgrade`, которая обновляет ее до новой версии. Это довольно удобно при выходе новых обновлений!

Но обычно файлы *cloud-config* будут создаваться автоматически, когда вы освоитесь с *RancherOS*, после чего их нужно загрузить в `ros` (для импорта используется параметр `-i`). Настраиваются параметров много; все они описаны на <http://docs.rancher.com>.

между контейнерами локально (между различными хостами) или, при необходимости, в другой облачной инфраструктуре (для чего применяются туннели IPsec). *Rancher* предоставляет обнаружение сервисов через DNS, благодаря которому контейнеры могут сообщить о своей доступности при запуске системы. Встроена также функция балансировки нагрузки на базе *NAppoxy* (одной из моих любимых открытых программ), которая своей простотой настройки и использования очень напоминает мне эластичный балансировщик нагрузки от Amazon. Тот, в свою очередь, использует средства мониторинга работоспособности контейнеров, способные выполнять задачи по обслуживанию в стиле *Kubernetes*, гарантируя, что в любой момент запущено нужное количество работоспособных сервисов. Это включает обычные правила подобия и различия (не запускайте все сервисы одного типа на одном хосте!). А самое главное — в платформу встроены плавающие релизы, и *Rancher* сам разворачивает новые релизы программного обеспечения и перенаправляет трафик на них. Он также откатывает изменения, если в новых релизах обнаруживаются ошибки. За последнее десятилетие я потратил уйму времени на такие вещи с разной степенью автоматизации (а порой и без нее), и эта функция для меня — просто бомба! Наконец, *Rancher* предлагает возможность управления хранилищем, включая создание снимков томов. По моему опыту, с управлением томами в *Docker* у команд разработчиков часто бывают сложности из-за требования персистентности данных, то есть это штука полезная. И всё это можно делать через web-интерфейс или в командной строке! Пора запустить всё это и попробовать.

» К счастью, добавить пользователей в *Rancher* проще простого. Доступны несколько видов аутентификации, включая любимый корпоративными пользователями Active Directory.

## Rancher в Vagrant и в RancherOS...

В такие времена я понимаю, что операционные системы стали похожи на «плоский мир» великого Терри Пратчетта, увы, недавно умершего. Вместо диска, расположенного на спине у слонов, стоящих на черепахе, мы воспользуемся *Vagrant* для запуска *RancherOS*, а затем воспользуемся его службами *Docker* для запуска *Rancher*. Надеюсь, аналогия была более-менее удачной. Если нет,

бояться нечего: мы клонируем репозиторий (репозитории) *Vagrant*, предоставленные разработчиками (чтобы это работало, *Vagrant* нужно устанавливать, например, вместе с *VirtualBox*). Инструкции см. на [www.vagrantup.com](http://www.vagrantup.com).

```
$ git clone https://github.com/rancher/os-vagrant.git  
$ cd os-vagrant  
$ vagrant up  
$ vagrant ssh
```

После этого мы подключимся к новой виртуальной машине *gancheros*. *RancherOS* поддерживает только пользователя с именем *rancher*. Мы можем взглянуть на вышеупомянутые системные процессы *Docker*, выполнив команды

```
$ sudo system-docker ps  
$ sudo system-docker images  
$ ps aux | grep docker
```

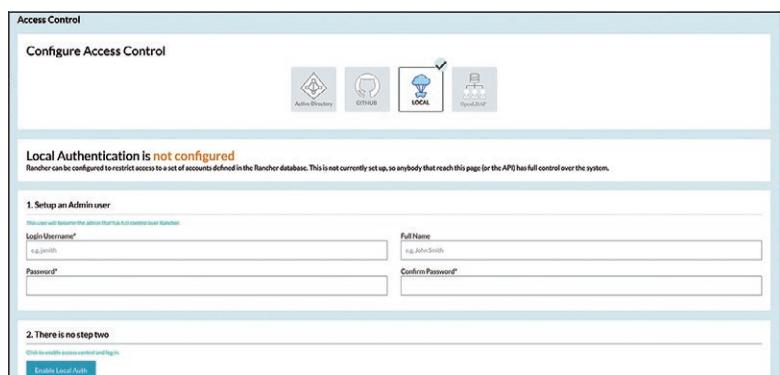
Они покажут запущенные по умолчанию контейнеры (например, *ntp*, *syslog*) и выведут список всех образов, доступных для использования ОС при необходимости (во время загрузки). Команда *ps* покажет три процесса, имеющие отношение к *Docker*, хотя они могут немного озадачить. Но не волнуйтесь, *user-docker* — это просто родительский процесс демона *none* *system-docker* *docker* (попробуйте произнести это после нескольких кружек пива).

Запустить *Rancher* на нашей виртуальной машине *RancherOS* нетрудно (хотя и загрузка, инициированная следующей командой, займет некоторое время):

```
$ sudo docker run -d --restart=always -p 8080:8080 rancher/server
```

Определите IP-адрес, назначенный вашей виртуальной машине *Vagrant* (командой *\$ ip addr show*), и затем откройте его в браузере с портом 8080. Вы увидите начальное окно *Rancher*. Обратите внимание на восклицательный знак над меню аудита. Он указывает, что по умолчанию в системе разрешен доступ без аутентификации. Поэтому первым делом я установлю желаемые параметры аутентификации. К счастью (этот вопрос всегда задают корпоративные пользователи), *Rancher* поддерживает Active Directory и OpenLDAP в качестве внешних опций контроля доступа. Также можно воспользоваться GitHub, что для некоторых групп разработчиков будет более естественным выбором. Можно настроить и старый добрый локальный доступ, что я и сделаю. На выбор этого варианта и добавление пользователя (первый пользователь по умолчанию становится администратором) должна уйти пара секунд.

В нижней правой части окна есть меню Download CLI [Загрузить клиент для командной строки]. Это удобно, и при выборе этого меню открываются варианты для Linux, Windows и OS X. При исключении двух недостойных и выборе единственной правильной операционной системы на мой локальный рабочий стол загружается утилита *rancher-compose*. Я воспользуюсь ею через месяц, когда покажу вам инфраструктуру на основе *Rancher*, обновления служб в реальном времени и многое другое!



## Как добавить свой первый хост?

Предваряя следующий урок, я хочу быстро добавить в мою инфраструктуру хост с самим *Rancher*. По нажатию кнопки Add Host [Добавить хост] откроется список доступных облачных провайдеров, но для этой локальной конфигурации я выберу Custom [Пользовательский]. Если, настроив доступ, я не выйду из системы и вернусь в интерфейс пользователя, может появиться предупреждение о достижимости IP-адреса; здесь его можно пригнорировать. На хосте, который я хочу добавить к своей инфраструктуре *Rancher*, надо из интерфейса пользователя скомандовать... в моем случае —

```
sudo docker run -d --privileged -v /var/run/docker.sock:/var/run/docker.sock rancher/agent:v0.8.2 http://
```

```
172.19.8.101:8080/v1/scripts/<extremely long string>
```

Однако в документации к *Rancher* указано, что мне следует добавить к этой команде `-e CATTLE_AGENT_IP=<server_ip>` — то есть для команды выше строка будет такой: `-e CATTLE_AGENT_IP=172.19.8.101`.

Я запустил эту команду в своем сеансе SSH *Vagrant*, и после ее выполнения компьютер извлек контейнер *Docker* с именем *rancher/agent*. После чего я нажал Close [Закрыть] в web-интерфейсе,

самую малость подождал, и хост появился в моем списке.

‘Cattle [Скот]’ — это отсылка к термину «Домашние животные и скот [Pets and Cattle]» из эпохи облачных вычислений, который разделяет системы на две категории. Pets — это системы, которые при их «заболевании» вы «лечите». Cattle — системы, которые проще убить и заменить новыми (и в эту категорию попадают большинство контейнеров). *Rancher* [хозяин ранчо] — неплохое имя для системы, управляющей огромным количеством одноразовых контейнеров.

# Let's Encrypt — SSL для всех

Зашифруем всё подряд! Защитите трафик на свой сайт бесплатным SSL-сертификатом и сделайте это из командной строки!

**A**нонсированный в конце 2014 г. проект *Let's Encrypt* в декабре 2015 г. наконец перешел в стадию бета, и это означает, что для его использования больше не нужны приглашения. Это «бесплатный, автоматизированный и открытый центр сертификации (CA)», который запускается исключительно для пользы общества.

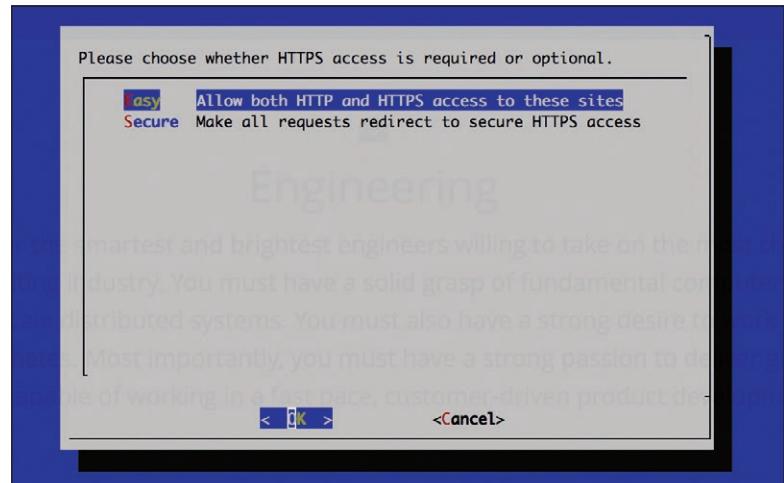
Поддерживаемый среди прочих Linux Foundation, данный проект кажется фантастическим предложением — ведь это бесплатные доверенные сертификаты (все, кто раньше платил за них сотни фунтов, поморщатся). А главное, проект обещает избавление от бюрократии при получении сертификата — вы когда-нибудь пробовали получить сертификат, работая в большой компании? Идея состоит в том, что такой центр сертификации будет совместной инициативой, от которой выиграет более широкое интернет-сообщество и которая при этом не попадет под контроль ни одной организации (по аналогии с интернет-протоколами нижнего уровня).

Следует признать, что сегодня отсутствие SSL на сайте выглядит дилетантски — поэтому вас может немного удивить то, что у меня есть несколько таких сайтов. Некоторые из них — небольшие персональные сайты, которые я содержу для других за так, другие — сайты сообществ, которые я содержу за так (пора задуматься, не многовато ли я делаю за так). Но как это работает?

Хотя установка сертификата для web-сервера никогда не была слишком сложной задачей, она включает, например, изменение директив *Apache*. Оказывается, не всем нравится редактировать текстовые файлы с данной целью (этакое кощунство!). Большинство людей просто хочет, чтобы что-то работало. В проекте этот процесс упрощается до запуска нескольких комманд: кроме того, возможно автоматически обновлять сертификаты таким же образом. (Другие сисадмины, пожалуйста, поднимите руки те из вас, кого никогда не заставало врасплох сообщение об истечении срока действия сертификата? Ах, вы? Врете.)

## Как это работает

В общих чертах, задача состоит в том, чтобы web-сервер доказал центру сертификации *Let's Encrypt*, что на нем находится тот сайт и на том домене, для которого был направлен запрос. Для этого на web-сервере запускается агент, который связывается с центром



» Несколько параметров *Curses* — всё, что вам нужно, чтобы защитить трафик своих посетителей от перехвата.

сертификации с помощью криптографии с открытым ключом. После этого центр сертификации просит агента, например, поместить файл туда, где он будет доступен центру сертификации, и выполняет несколько запросов подписи. После их подтверждения агент может запросить сертификат и установить его.

В будущем ожидается выпуск пакетов для различных дистрибутивов, а сейчас доступен репозиторий *letsencrypt*. Первый шаг — клонировать репозиторий git и запустить комманду *letsencrypt-auto*:

```
$ git clone https://github.com/letsencrypt/letsencrypt
$ cd letsencrypt
$ ./letsencrypt-auto --apache -d <your domain name>
```

Эта команда настроила для меня различные зависимости (запросив пароль sudo), что заняло некоторое время. Было установлено довольно много пакетов (*openssl* и ему подобные), поэтому протестируйте это, если работаете в среде, где добавление пакетов может вызвать проблемы. На этапе настройки я также получил несколько предупреждений (из-за отсутствия некоторых библиотек, которые бы сделали TLS более безопасным).

Затем надо настроить несколько параметров через интерфейс *Curses* (контактный адрес электронной почты и включение обоих протоколов HTTP и HTTPS), и скрипт завершится, записав несколько ключей в локальный каталог (их определено стоит сохранить, так как они потребуются для обновлений). Моя конфигурация *Apache* обновила пути к новым сертификатам, и когда я открыл свой тестовый сайт, в адресной строке браузера появился замок. Получилось!

Испортив файлы конфигурации (во многом из-за скверного знания *Apache*), я ухитрился сломать свой сайт, но повторно запустил комманду *letsencrypt-auto* и смог восстановить свои настройки (таким же образом можно восстановить сертификат, причем количество восстановлений в неделю ограничено). Теперь я собираюсь защитить все свои сайты. Возможно, всем это делать со своими сайтами не обязательно, но *Let's Encrypt* — подлинное достижение.

» *Let's Encrypt* — проект, миссией которого является защита трафика по всему Интернету с сертификатами, доступными по королевской цене в £ 0,00.

# 7 советов быстрого поиска работы от hh.ru

- 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».
- Узнайте о своих способностях.** Вы все еще в поиске своего призыва? Пройдите онлайн-тест «Профориентация»\* [hh.ru/article/proforientation\\_promo](http://hh.ru/article/proforientation_promo) и узнайте, какая работа вам больше всего подойдет.
- 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.
- 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.
- Действуйте.** Откликайтесь на все интересующие вас вакансии. Пишите сопроводительные письма работодателю, поясня, почему вас интересует эта вакансия.
- Сделайте резюме заметным.** [hh.ru/applicant/services](http://hh.ru/applicant/services) Подключите «Яркое резюме»\*, чтобы выделить резюме цветом, и «Автообновление»\*, чтобы поднимать его в результатах поиска. Работодатели обратят на вас внимание.
- 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!**

# Учебники

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН  
Подтвердивший  
свою квалификацию  
физик.

## ПК вам в помощь

Я могу рисовать, как Рафаэль, но мне понадобится вся жизнь, чтобы научиться рисовать так, как рисует ребенок  
Пабло Пикассо

**М**ало кто из людей умеет хорошо рисовать. В смысле — рисовать так, чтобы на это смотрели с умилением отнюдь не только близкие и одновременно любящие родственники. Похоже, у криворукого большинства на улице праздник — или хотя бы потенциальный повод помечтать.

Алекс Шампандар [Alex Champandard] выложил на Github всего-то 500-строчный скрипт *doodle.py* (пакет *neural-doodle*, лицензия типа BSD). Теперь любой умеющий рисовать в графическом редакторе «каракули», применяя три-четыре цвета, может преобразовать их во вполне себе художественные произведения. Особенно неплохо получается прикинуться импрессионистом. Алгоритм изложен в статье *Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artwork*, выложенной в доступ на [arxiv.org](http://arxiv.org) под №1603.01768.

Естественно, это не значит, что художникам нужно срочно менять профессию. Зато появляется новая возможность создать приятные для глаза цветовые пятна, теперь доступная более широкому числу людей, включая и художников. Со временем число подобных инструментов будет расти, но это не значит, что компьютер может заменить человека абсолютно, хоть в го он его уже и обыграл. Должен же кто-нибудь рисовать «каракули». E.m.Baldin@inp.nsk.su

## В этом месяце вы научитесь...



### Сортировать строки ..... 52

Порядок должен быть во всем! Но тогда надо это всё рассортировать, чем и занялся **Дмитрий Пантелейчиков** с помощью ключевых утилит.



### Прикладываться к иконкам ..... 56

Ник Пирс перемещает и масштабирует иконки всяческими способами, чтобы взглянув на экран сразу выхватывал самое главное.



### Загружать компьютер ..... 58

Вы нажали кнопку Пуск, собираясь поработать, и... ничего не происходит. Скорее обращайтесь к **Нику Пирсу**! Он разберется с этой неприятностью.



### Ставить пакеты ..... 62

Дистрибутив со скользящими релизами, конечно, обновляет сам себя, но **Нейлу Ботвику** этого мало — он добывает пакеты через *Portage*.



### Узнавать подробности .. 64

Как получить данные о неисправной системе, если она защищена паролем? Утилита **Джеймса Ханта** тактично обо всем повыспросит.



### Анализировать данные ..... 68

Мы говорим — Большие Данные, подразумеваем — Hadoop. **Рафик Рехман** знакомит вас с этой знаменитой системой.



### Работать с шардами ..... 72

Что такое «шард», вам пора знать из предыдущих номеров журнала (и в этом номере тоже есть пояснение), а **Лада Шерышова** расскажет дальнейшее.



### Набирать номера ..... 76

Максим Черепанов завершает серию своих статей по IP-телефонии подготовкой диалплана и тренингом вежливого автоответчика.

## АКАДЕМИЯ КОДИНГА



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

### Скоренько о Swift ..... 80

Краткое введение в Swift и его возможности от **Михаила Цукалоса** даст вам представление об этом открытом языке программирования.

### Управляем MongoDB ..... 84

**Михалис Цукалос** обучает администрированию распределенных баз данных, попутно создавая их реплики.

ЧАСТЬ 5

# GNU Core Utilities

Дмитрий Пантелейчев продолжает постигать основы системы GNU/Linux через ее основополагающий пакет программ.



Наш  
эксперт

Дмитрий Пантелейчев считает, что любую технологию надо изучать от простого к сложному, и каждый шаг закреплять практическими примерами.

**C**егодня мы рассмотрим программы, отвечающие за сортировку строк — это утилиты *sort*, *shuf*, *uniq*, *comm*, *ptx*, *tsopt*; а также программу *tee*, которая выводит поступившую информацию в несколько потоков.

## sort

Само слово *sort* интуитивно наводит на мысль о сортировке — в данном случае, текстовых строк. Если мы вызовем эту команду, подставив ей многострочный текст, то строки этого текста будут отсортированы по алфавиту. Давайте подставим перечень содержимого столового набора. Напомню: оператор << означает, что чтение прекращается при вводе символов, указанных после <<.

```
sort << EOF
Ложка
Вилка
Нож
EOF
```

После ввода этой команды на экране должна появиться «сервировка», отсортированная по алфавиту.

```
Вилка
Ложка
Нож
```

Как и любая другая команда, выводящая какую-либо информацию, *sort* выполняет вывод в стандартный поток *stdout*, который по умолчанию подключен к эмулятору терминала; но вывод можно направить и на другое устройство. Например, в файл:

```
sort > tableset << EOF
Ложка
Вилка
Нож
EOF
```

В вашем рабочем каталоге появится новый файл — *tableset*, что покажут команды *ls* или *dir*. Если вывести содержимое этого файла командой *cat tableset*, оно появится на экране по алфавиту.

Откуда же система знает алфавитный порядок символов?

Алфавитный порядок символов определяется через набор параметров под общим названием «локаль». Локаль, или культурная

среда — это совокупность параметров, характеризующих культурное и географическое окружение пользователя: национальный алфавит, формат отображения даты и времени, названия месяцев и многое другое. Локаль вашей системы покажет утилита *locale* (она не входит в *coreutils* и устанавливается с пакетом *glibc*). Результат команды *locale* должен выглядеть как на рис. 1.

В этом списке первой отображается переменная окружения *LANG*, содержащая идентификатор текущей локали системы. Следующая переменная, *LANGUAGE*, необязательна. Она используется, если пользователь хочет указать, на каком языке получать сообщения, не переведенные на язык локали. В *LANGUAGE* указывается приоритет языков. Так, если шведский пользователь хочет получать сообщения на немецком языке, если они недоступны на шведском, он задает *LANGUAGE* значение: 'sv:de'. Это означает «сначала, если можно, шведский; затем немецкий».

Остальные переменные — категории локали. Значение каждой из этих переменных — название локали. Разные категории могут использовать разные локали. Категории имеют такие значения:

- » *LC\_IDENTIFICATION* — уникальный идентификатор локали.
- » *LC\_CTYPE* — информация о наборе символов национального языка, их классификация и правила преобразований.
- » *LC\_COLLATE* — алфавитный порядок букв для сортировки строк.
- » *LC\_TIME* — формат даты и времени.
- » *LC\_NUMERIC* — формат чисел (например, отображение десятичной дроби).
- » *LC\_MONETARY* — формат денежных величин.
- » *LC\_MESSAGES* — сообщения пользовательского интерфейса на национальном языке.
- » *LC\_PAPER* — информация о стандартном размере бумаги для печати документов (например, A4).
- » *LC\_MEASUREMENT* — стандартная система измерений, принятая в культуре пользователя (например, метрическая).
- » *LC\_NAME* — формат личного имени человека (например, имя/отчество/фамилия).
- » *LC\_ADDRESS* — формат почтового адреса.
- » *LC\_TELEPHONE* — формат телефонного номера.
- » *LC\_ALL* — будучи заданным, значение этой переменной перекрывает значения всех вышеуказанных категорий, приводя их все к единой локали. Она используется в скриптах, для которых важно, чтобы их выполнение не зависело от культуры системы.

Команда *sort* с ключом *-c* (полная версия — *--check*) позволяет проверить, правильно ли отсортирован список строк. Давайте создадим новый файл, со списком времен года. Введем их названия в порядке следования:

```
cat > seasons << EOF
Зима
Весна
Лето
Осень
EOF
```

А как на это посмотрит команда *sort*? Давайте проверим:  
*sort -c seasons*  
*sort: seasons:2: неправильный порядок: Весна*

```
dima@dima-System-Product-Name ~ $ locale
LANG=ru_RU.UTF-8
LANGUAGE=
LC_CTYPE="ru_RU.UTF-8"
LC_NUMERIC="ru_RU.UTF-8"
LC_TIME="ru_RU.UTF-8"
LC_COLLATE="ru_RU.UTF-8"
LC_MONETARY="ru_RU.UTF-8"
LC_MESSAGES="ru_RU.UTF-8"
LC_PAPER="ru_RU.UTF-8"
LC_NAME="ru_RU.UTF-8"
LC_ADDRESS="ru_RU.UTF-8"
LC_TELEPHONE="ru_RU.UTF-8"
LC_MEASUREMENT="ru_RU.UTF-8"
LC_IDENTIFICATION="ru_RU.UTF-8"
LC_ALL=
```

» Рис. 1. Информация о локали.

Ошибочка вышла: ведь по алфавиту Весна раньше Зимы.

Вообще команда `sort` имеет необычайно много ключей, о которых можно прочитать, обратившись к справочным системам по команде `man sort` или `info sort`. Можно указать, принимать во внимание или нет начальные пробелы (`-b`), различать буквы по регистрам или нет (`-f`), игнорировать непечатаемые символы или нет (`-i`), и др.

Например, как отсортировать строки, начинающиеся с “1”, “2” и “10”? Если сортировать строго по символам, то “10” идет раньше, чем “2”, а “10.” раньше, чем “1.”. Но часто бывает нужно выделить числовую часть строк и отсортировать их в числовом порядке. Программа может реализовать оба варианта сортировки. Создадим файл `book-index`.

```
cat > book-index << EOF
```

10. Заключение

2. Обзор литературы

1. Введение

EOF

Сортировка по умолчанию не учитывает числовые значения символов, поэтому даст нежелательный результат:

```
sort book-index
```

10. Заключение

1. Введение

2. Обзор литературы

Но добавление ключа `-n` (или `--numeric-sort`, или `--sort=numeric`) поможет отсортировать эти строки правильно:

```
sort -n book-index
```

1. Введение

2. Обзор литературы

10. Заключение

## shuf

Теперь займемся утилитой `shuf`. Ее роль совершенно противоположная: она перемешивает строки в случайном порядке. Посмотрите, например, как она выводит список времен года, который находится в созданном нами файле `seasons`.

```
shuf seasons
```

Введя эту команду неоднократно, вы каждый раз увидите новый результат.

Ключ `-n` позволяет задать количество строк, которое надо вывести (без него по умолчанию выводятся все строки). Это бывает удобно, когда исходный список очень большого размера.

```
shuf -n 2 seasons
```

Или даже можно вывести один случайный элемент из списка.

```
echo Сегодня мы рисуем пейзаж на тему `shuf -n 1 seasons`
```

Сегодня мы рисуем пейзаж на тему Осень

Здесь мы забежали вперед, употребив пока не знакомую утилиту — `echo`. Но пусть символы ` вас не пугают. Потом узнаете, что они означают.

Кстати, насчет `echo`. Ключ `-e` (`--echo`) позволяет задавать исходный список сразу после команды в виде списка параметров через пробелы. Параметры из нескольких слов вводятся в кавычках:

```
shuf -e Коля "Саша Иванов" "Саша Захаров"
```

Коля

Саша Захаров

Саша Иванов

Ключ `-i` позволяет вывести в случайном порядке набор целых чисел. Наименьшее и наибольшее из этих чисел должны быть указаны после ключа.

```
shuf -i 15-20
```

16

17

18

20

19

15

## uniq

Утилита `uniq` преобразует повторяющиеся соседние строчки в одну строчку. Создадим файл `months` с повторяющейся строкой:

```
cat > months << EOF
```

Январь

Февраль

Февраль

Март

Апрель

EOF

Откроем его программой `uniq`. Повторяющиеся строки объединятся в одну:

```
uniq months
```

Январь

Февраль

Март

Апрель

Добавим ключ `-c`, мы узнаем число повторов каждой строки:

```
uniq -c months
```

1 Январь

2 Февраль

1 Март

1 Апрель

Ключ `-d` исключает из результата уникальные строки:

```
uniq -d months
```

Февраль

А ключ `-f` указывает, сколько полей следует пропустить, прежде чем оставшаяся часть строки будет сравниваться с аналогичными оставшимися частями других строк. Поле — это часть строки, отделенная пробелами или табуляцией.

Давайте для иллюстрации придумаем сложный пример, в котором будут задействованы сразу три утилиты: `sort`, `uniq` и `cut`. Перечислим несколько городов мира с указанием страны и количества жителей. Затем выведем только список стран, которые присутствовали в исходном списке.

Для начала создадим файл с исходным списком.

```
cat > cities << EOF
```

Шанхай Китай 17836133

Мумбаи Индия 12478447

Карачи Пакистан 13205339

Пекин Китай 11716620

Гуанчжоу Китай 11070654

Дели Индия 11007835

Лахор Пакистан 10520000

Шэнъянь Китай 10467400

EOF

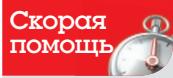
Теперь выведем уникальный список только стран:

```
sort -k 2.1 cities | uniq -f 1 -w 5 | cut -f 2 -d "
```

Проведем разбор этой команды. Ключ `-K` со значением `2.1` означает, что сортировать строки надо не с начала строки, а по первому символу второго поля. Поскольку в каждой строке название страны стоит после названия города и отделено от него пробелом, то название страны и есть поле номер два, и по нему идет сортировка. Далее идет параметр, указывающий источник входящего текста. В данном случае указано имя файла, содержимое которого сортируется. Но можно получить исходные данные и из другого места. Например, если вместо имени файла ввести `<< EOF`, то входные данные будут вводиться в программу с клавиатуры, пока не встретится сочетание `EOF`, после чего результат работы программы сразу будет выведен на экран.

Вторая команда `uniq` отделена от первой знаком `|`. Это хорошо нам известный конвейер: результат работы первой программы поступает в качестве исходных данных во вторую программу. Обратите внимание, что команда `uniq` здесь не имеет параметра, указывающего на входящие данные, а имеет только ключи. Это потому,

»



Технология «подстановки процесса» позволяет на место любого входящего параметра подставить вызов программы, выводящей данные. А также, наоборот, исходящий поток направить на вход любой программы.

что на входящие данные указывает символ конвейера `|`, стоящий перед ней.

Ключ `-f` команды `unzip` показывает, сколько полей надо пропустить, прежде, чем проверять строки на дублирование. Поскольку мы проверяем страны, то надо пропустить название города, т.е. одно поле. Ключ `-W` указывает, сколько символов надо проверять на дублирование, чтобы не просматривать всю строку до конца. Для нашего примера достаточно будет сравнивать пять первых символов.

Утилита `cut` осуществляет частичный вывод исходных строк. Поскольку мы выводим только страны, т.е. только второе поле, на что указывает ключ `-f` со значением 2. По умолчанию утилита `cut` считает полями только части строк, разделенные табуляцией. Но здесь у нас поля разделяются пробелами, и это надо особо указать с помощью ключа `-d` со значением `" "`.

Результат должен получиться таким:

```
Индия  
Китай  
Пакистан
```

## comm

Утилита `comm` сравнивает два отсортированных набора строк. Но давайте для эксперимента подставим ей два неотсортированных файла, чтобы посмотреть, что получится. Создадим два файла — `digits1` и `digits2`, с наборами чисел в произвольном порядке:

```
cat > digits1 << EOF  
4  
2  
1  
3  
EOF  
cat > digits2 << EOF  
3  
0  
2  
1  
EOF
```

Попробуем сравнить эти два файла с помощью команды `comm`:

```
comm digits1 digits2
```

Программа, естественно, отругает нас, выдав два сообщения: «данные файла 1 не отсортированы» и «данные файла 2 не отсортированы». Конечно, мы бы могли отсортировать эти наборы цифр, записать результаты в другие файлы и новые файлы сравнить. Но нельзя ли как-нибудь обойтись без лишних файлов?

Вам уже знакома технология конвейера, когда выход одной программы поступает на вход другой программы. Но в данном случае наша программа имеет два входа, поэтому технологию конвейера применить не получится.

К счастью, мы можем воспользоваться другой технологией: подстановкой процесса [process substitution]. Суть ее в том, что на место параметра, где должен быть указан файл, подставляется вызов команды, заключенный в скобки с добавлением знака направления потока. Между этим знаком и скобкой не должно быть пробела. В данном случае мы вызовем программу `sort`, результат работы которой и подставится в качестве параметра в программу `comm`:

```
comm <(sort digits1) <(sort digits2)
```

Вот теперь видно, что цифра 4 есть только в первом наборе, цифра 0 — только во втором, а цифры 1, 2, 3 — в обоих наборах.

```
dima@dima-System-Product-Name ~ $ comm <(sort digits1) <(sort digits2)  
0  
1  
2  
3  
4 dima@dima-System-Product-Name ~ $
```

```
dima@dima-System-Product-Name ~ $ ptx love-story  
great a love can be/ Where do  
she brings to me Where should  
/of how great a love can be  
/that is older than the sea  
love can be/ Where do I begin  
of how great a love can be/  
about the love she brings to me  
/To tell the story of how great  
/than the sea The simple truth  
/story of how great a love can  
great a love can/ Where do I  
/simple truth about the love she  
/the story of how great a love  
how great a love can/ Where  
/begin To tell the story of how  
/do I begin To tell the story of  
/be The sweet love story that  
/tell the story of how great a  
/great a love can be The sweet  
/sea The simple truth about the  
about the love she brings to me  
/do I begin To tell the story  
/be The sweet love story that is
```

```
I begin To tell the story of how  
I start? /the love  
The sweet love story that is older/  
The simple truth about the love/  
To tell the story of how great a  
Where do I begin To tell the story  
Where should I start? /truth  
a love can be The sweet love story/  
about the love she brings to me/  
be The sweet love story that is/  
begin To tell the story of how  
brings to me Where should I start?  
can be The sweet love story that/  
do I begin To tell the story of  
great a love can be The sweet love/  
is older than the sea The simple/  
love can be The sweet love story/  
love story that is older than the/  
love she brings to me Where should/  
me Where should I start? /truth  
of how great a love can be The/  
older than the sea The simple/
```

Рис. 3. Индекс слов с контекстом.

## ptx

`ptx` формирует упорядоченный список слов, входящих в какой-нибудь текст, и отображает вместе с их контекстом. Увы, пока она умеет работать только с символами из кодовой таблицы ISO 8859-1 (то есть из первых 256 кодовых позиций Unicode). Сюда входят только символы западноевропейских языков, и работать с русским языком программа не сможет.

Создадим файл `love-story` с начальными строками текста одной популярной и очень красивой английской песни:

```
cat > love-story << EOF  
Where do I begin  
To tell the story of how great a love can be  
The sweet love story that is older than the sea  
The simple truth about the love she brings to me  
Where should I start?  
EOF
```

Давайте посмотрим, как программа `ptx` это обработает.

```
ptx love-story
```

На рис. 3 мы можем видеть, что слова из индекса расположены сверху вниз, окружены контекстом (т.е. теми словами, которыми они окружены в исходном тексте), и отсортированы по алфавиту с учетом регистра: сначала идут слова, начинающиеся с заглавных букв, а затем слова, начинающиеся со строчных.

Добавим к этой команде два ключа:

```
ptx -fA love-story
```

Ключ `-f` укажет, что сортировка индексных слов должна быть без учета регистра. Ключ `-A` потребует в начало каждой строки добавлять имя файла и номер строки, где это слово с таким контекстом встречается. Результат — на рис. 4. Как обычно, результат можно выводить не только на экран, но и в файл:

```
ptx -fA love-story > love-story-index
```

По умолчанию `ptx` индексирует каждое слово. Но можно велеть ей индексировать только слова, соответствующие какому-либо регулярному выражению. Для этой цели используется ключ `-W`.

Создадим новый файл и пометим в нем звездочками слова, которые хотим проиндексировать:

```
cat > love-story-reg << EOF  
Where do I begin  
To tell the *story* of how great a *love* can be  
The *sweet* love story that is older than the sea  
The simple *truth* about the love she brings to me  
Where should I start?  
EOF
```

Теперь снова введем команду, добавив в нее ключ `-W`, и в качестве его значения укажем регулярное выражение, по которому ключевые слова будут выделены из всех прочих.

```
ptx -W '\*w+\*' -fA love-story-reg
```

В результате будут выведены только помеченные нами слова `love`, `story` и `truth` вместе с их контекстом.

Рис. 2. Результат работы команды `comm`.

Вообще программа эта довольно сложная и имеет много ключей. Почитайте про них в справочной системе.

## tsort

Утилита *tsort* вовлекает новый термин: «топологическая сортировка» слов. Для такой сортировки программа должна получить несколько параметров в виде пар слов, разделенных пробелом. Причем первое слово из каждой пары является иерархически вышестоящим по отношению ко второму. Но второе слово в другой паре может само быть вышестоящим и иметь своего «подчиненного». Например, понятие *software* — более высокого уровня, чем *os*, *office*, *game*, *browser*, потому что эти последние являются частными случаями понятия *software*. А понятия *windows*, *linux*, *freebsd*, *reactos* — частные случаи понятия *os*. Думаю, принцип ясен.

Программа *tsort* распознает всю эту «иерархию» и сортирует слова от высшего уровня к низшему. Это бывает нужно, например, чтобы определить порядок установки программных пакетов, для уверенности, что в системе уже есть все пакеты, от которых данный пакет зависит.

```
tsort << EOF
linux ubuntu
os windows
linux debian
os freebsd
software office
os linux
software os
office libreoffice
EOF
```

Результат должен получиться такой:

```
software
os
office
linux
freebsd
windows
libreoffice
debian
ubuntu
```

Как видите, иерархичность строго соблюдается: понятие более низкого уровня никогда не стоит раньше понятия более высокого уровня.

## tee

Вернемся снова к понятию «стандартный вывод». Как мы знаем, это зарезервированный поток для вывода данных. Задача программы *tee* в том, чтобы, помимо стандартного вывода, вывести поступившую информацию и в другие потоки. Посмотрим, как это выглядит:

```
tee << EOF helloworld1 helloworld2
Здравствуй, мир, здравствуй, друг,
Здравствуй, песен щедрый круг!
EOF
```

В этой команде первый параметр означает входящие данные, а все последующие параметры — имена исходящих файлов (их можно указать сколько угодно). Входящие данные здесь берутся с консоли (пока не будет введено *EOF*), а исходящие данные, во-первых, отправляются в стандартный вывод (на консоль), во-вторых, запишутся в файлы, указанные в параметрах. Сразу после ввода *EOF* в консоли должны появиться две введенные строчки песни, а в текущем каталоге — файлы *helloworld1* и *helloworld2*. Проверьте их содержимое с помощью команд *cat helloworld1* и *cat helloworld2*. В этих файлах должны быть те же самые две строчки.

Что делать, если мы на экран не хотим выводить ничего, а хотим результаты выводить только в файлы? Для этого, сразу же

```
dima@dim-a-System-Product-Name ~ $ ptx -f -A love-story
love-story:2: /the story of how great
love-story:4: /sea The simple truth
love-story:2: /of how great a love can
love-story:1: how great/ Where do I
love-story:4: I/ /about the love she
love-story:2: /of how great a love
love-story:1: story of how/ Where
love-story:2: /tell the story of how
love-story:2: /To tell the story of
love-story:1: of how/ Where do
love-story:2: to me Where should
love-story:3: /sweet love story that
love-story:2: /story of how great a
love-story:3: /a love can be The sweet
love-story:4: /simple truth about the
love-story:4: /the love she brings to
love-story:2: /begin To tell the story
love-story:3: /Love story that is older
love-story:3: /that is older than the
love-story:4: /truth about the love
love-story:5: she brings to me Where
love-story:4: /older than the sea The
love-story:5: to me Where should I
love-story:5: /she brings
```

**Рис. 4. Индекс слов с контекстом, именем файла и номером строки.**

после указания источника информации, мы должны указать имя еще одного файла, поставив перед ним символ *>* (перенаправление стандартного вывода). Тогда то, что предназначалось для консоли, уйдет в файл *helloworld3*.

```
tee << EOF > helloworld3 helloworld1 helloworld2
Здравствуй, мир, здравствуй, друг,
Здравствуй, песен щедрый круг!
EOF
```

Теперь у нас появилось желание в каждом из этих файлов добавить две новых строчки, сохранив и те, которые уже имеются. Мы знаем, что перенаправление стандартного потока в файл с сохранением старого содержимого файла осуществляется с помощью оператора *>>*. Поэтому первая мысль возникает сделать так.

```
tee << EOF >> helloworld1 helloworld2 helloworld3
Здравствуй, миг, здравствуй, век,
Здравствуй, добрый человек!
EOF
```

Не спешите это вводить! Строчки добавятся к старому содержимому только в файле *helloworld1*. А в двух других файлах они полностью заменят старое содержимое. Ведь оператор *>>* относится только к стандартному выводу, который без него выводил бы информацию в консоль. Остальные параметры просто обычным образом создают новый файл или перезаписывают имеющийся.

Но на самом деле решение есть. Дело в том, что программы выводят информацию, строго говоря, не в файлы, а в потоки. Просто эти потоки по умолчанию направляются в файлы. Но их можно направить и в другое место.

В этом уроке уже шла речь о технологии «подстановки процесса». Мы рассмотрели ситуацию, когда команда ожидает два входящих потока, и на место этих двух входящих потоков, вместо имен файлов, можно подставить исходящие данные других программ: *comm <(sort digits1) <(sort digits2)*

Точно также можно поступить и с исходящими потоками. Вместо файлов можно вывести потоки на входы других программ, в данном случае — на вход программы *cat*. Синтаксис команды будет такой же, только оператор направления смотрит в другую сторону:

```
tee << EOF >> helloworld1 >(cat >> helloworld2) >(cat >>
helloworld3)
Здравствуй, миг, здравствуй, век,
Здравствуй, добрый человек!
EOF
```

Проверьте. Во всех трех файлах новые строчки должны присоединяться к старым строчкам.

До встречи на следующем уроке! [LXF](#)

```
dima@dim-a-System-Product-Name ~ $ ptx -W '\*\w\*\*' -f -A love-story-red
love-story-reg:2: /of how great a *love* can be The/
love-story-reg:2: /a begin To tell the *story* of how great a/
love-story-reg:3: /t a *love* can be The *sweet* love story that/
love-story-reg:4: /an the sea The simple *truth* about the love s/
dima@dim-a-System-Product-Name ~ $
```

**Рис. 5. Индексация ключевых слов.**

# Ubuntu: Файлы, папки & иконки

**Ник Пирс** изучает, как удобно разместить ярлыки приложения, файла и папки на рабочем столе Unity.



Наш эксперт

**Ник Пирс** возится с компьютерами более 30 лет, со времен BBC Micro и Dragon 32. Но его навыки программирования остаются недоразвитыми.

Рабочий стол — подходящее место для ярлыков к любимым приложениям (известных как средства запуска — launcher), системных инструментов и даже папок и файлов. При желании вы даже можете хранить прямо на рабочем столе файлы и папки, хотя обычно это не самая хорошая практика — разве что как кратковременная мера.

На этом уроке мы планируем подробно разобрать, как работают иконки рабочего стола, и выяснить, как добавлять, удалять и управлять иконками, которые вы туда поместили. Мы также покажем вам, как индивидуализировать ваши иконки, сделав их более (или менее) заметными, включая замену или редактирование самой иконки. В данном руководстве мы будем ориентироваться на рабочий стол Unity (см. врезку «На других рабочих столах» вверху на соседней странице для советов по работе с другими средами).

## Анатомия файла рабочего стола

У каждого пользователя есть свой собственный рабочий стол, содержимое которого хранится в папке `/home/Desktop`. Скопируйте в нее файл или папку, и они также появятся на рабочем столе — щелчок по файлу откроет его, а щелчок по папке откроет ее содержимое в менеджере файлов, как обычно. Все то, что хранится на вашем рабочем столе, является личным и частным — оно не появится на чьем-либо еще рабочем столе.

Ярлыки рабочего стола — это несколько иное. Они появляются с такой же иконкой, как и родительский элемент — файл, папка или приложение — с которым они связаны. Ярлыки файла и папки определяются по маленькой стрелке вверх, появляющейся в нижнем правом углу. Также они по умолчанию получают имя «Ссылка на <имя файла>», хотя вы легко можете переименовать их



Создавайте новые ярлыки с нуля с помощью инструмента изготовления ярлыков приложений Main Menu.

по своему вкусу, через меню Properties [Свойства]. А вот иконки приложений не имеют подобного знака отличия — вместо этого они принимают внешний вид родительского элемента вместе с иконкой, присвоенной тому.

По сути, эти файлы — это простые текстовые файлы, которым дано расширение файла `.desktop` (оно не отображается на рабочем столе или в вашем менеджере файлов — перейдите в `/home/Desktop` в окне терминала и используйте `ls`, чтобы просмотреть их). Они определяют различные атрибуты ярлыка с помощью следующего синтаксиса:

```
[Desktop Entry]
Version=x.y
Name=ИмяПрограммы
Comment=Мой комментарий
Exec=/home/alex/Documents/exec.sh
Icon=/home/alex/Pictures/icon.png
Terminal=false
Type=Application
Categories=Utility;Application;
```

Каждый атрибут самоочевиден, но их создание в текстовом редакторе — весьма сложная задача (и по большей части ненужная). Для полного руководства посетите <http://bit.ly/UnityLaunchersAndDesktopFiles>, но обычно это процесс нудный и необязательный. Впервые установив Ubuntu, вы найдете свой рабочий стол абсолютно пустым, но также обнаружите множество уже созданных ярлыков — перейдя в `/usr/share/applications`, найдете ярлыки, указывающие на большую часть приложений и системных инструментов, уже установленных на вашем ПК. Щелкните по одному из них правой кнопкой мыши и выберите Properties, чтобы увидеть, куда он указывает. Если вы попытаетесь перетащить один из ярлыков этих приложений на рабочий стол, вы получите ошибку

## Измените набор иконок

Если вы ищете простой и быстрый способ изменить ваши иконки, рассмотрите вариант перехода на другой набор иконок, который заменит многие — если не все — ваши существующие иконки на новые, причем разработанные в едином стиле. Установите *Unity Tweak Tool* через *Software Center*, затем проведите поиск по сети по запросу 'icon set ubuntu', чтобы найти подходящий для установки, симпатичный вам набор иконок.

Например, чтобы установить набор иконок Moka, сделайте следующее:

```
$ sudo add-apt-repository ppa:moka/stable
$ sudo apt-get update
$ sudo apt-get install moka-icon-theme
```

После установки откройте *Unity Tweak Tool* и выберите Icons [Значки]. Вам предложат множество различных наборов — уже имеющиеся наборы не вносят радикальных изменений во внешний вид, но выберите Moka, и вы тут же почувствуете разницу. Вы также заметите, что изменение затронуло все ярлыки рабочего стола, которые вы создали с иконкой приложения по умолчанию.

## Другие рабочие столы

Если ваш рабочий стол — не Unity, вы обнаружите, что работа с иконками на нем несколько отличается. Для начинающих, большинство других рабочих столов включают удобный инструмент создания ярлыков прямо из контекстного меню рабочего стола — просто щелкните правой кнопкой мыши и выберите Create Launcher [Создать запуск], если вы используете Cinnamon/Xfce, или Create New Shortcut [Создать ярлык] в LXDE.

KDE Plasma требует, чтобы вы настроили всё через ее собственную программу запуска. Нажмите кнопку KDE в нижнем левом углу экрана, затем выясните расположение нужного приложения через всплывающее меню. Щелкните по нему правой кнопкой мыши и выберите Add to Desktop [Добавить на рабочий стол]. Создастся иконка виджета, который работает иным образом — но если вы знакомы

с другими виджетами Plasma, у вас не будет проблем с перемещением, изменением размера и даже вращением иконки с помощью панели инструментов виджета.

Если вы пользователь Gnome 3, иконки на рабочем столе не включены по умолчанию. Вам понадобится установить *Gnome Tweak Tool* через менеджер пакетов, затем запустить его из Utilities [Утилиты] и переключатель Icons on Desktop [Значки на рабочем столе] на On [Вкл]. После этого вы сможете вручную перетаскивать ярлыки из папки *usr/share/applications* на рабочий стол. Вы увидите ошибку насчет прав доступа к файлу — щелкните правой кнопкой по новому ярлыку и выберите вкладку Properties > Permissions [Разрешения]. Отметьте Allow executing file as Program [Разрешить



КDE Plasma рассматривает иконки рабочего стола как виджеты, которые можно перемещать, менять их размер и вращать.

использовать как приложение], и вы увидите появившуюся иконку, а ярлык теперь будет работать как положено.

прав доступа, а если вы очень постараетесь — нечто вроде ярлыка все же появится, но с неверной иконкой, и работать он не будет.

Есть три простых и быстрых способа создавать ярлыки. Если вы хотите поместить ярлык в существующую папку или файл (но не приложение), перейдите в его расположение с помощью менеджера файлов *Nautilus*, нажмите на нем правой кнопкой мыши и выберите Make Link [Сделать ссылку]. Появится ярлык — просто перетащите его на рабочий стол, затем щелкните по нему правой кнопкой мыши и выберите Properties, чтобы переименовать его.

Простейший способ создать ярлык для установленного приложения — в сочетании с Dash. Откройте его из Launcher, затем нажмите белую кнопку Maximise вверху слева, чтобы уменьшить размер. Напишите название приложения, которое вы ищете; просто перетащите его из Dash на рабочий стол туда, где вы хотите, чтобы он появился. Заметьте, что в некоторых исключительных случаях — одним из которых является Start Center в *LibreOffice* — вы можете обнаружить, что ярлык не появился корректно и выдает ошибку Untrusted application launcher [Недоверенный запуск приложения]. Обычно это происходит из-за того, что файл в */usr/share/applications* сам по себе является ссылкой на другой файл *.desktop* — вам надо найти исходную ссылку (*/usr/lib/libreoffice/share/xdg* в случае с *LibreOffice*) и перетащить ее оттуда. Сообщения об ошибке доступа игнорируйте — вы найдете копии ярлыка в полном порядке.

Если вы предпочли бы создавать ярлыки рабочего стола с нуля без привлечения текстового редактора, откройте Dash и запустите поиск по запросу 'alacarte', затем нажмите на появившуюся иконку Main Menu [Главное меню]. После этого нажмите на New [Создать]. Откроется новое диалоговое окно, позволяющее создать ваш ярлык с нуля: введите его имя, затем тип выбранной команды (отметьте Launch in Terminal? [Запуск в терминале?], если это возможно) или нажмите Browse [Просмотр], чтобы выбрать приложение с помощью диалогового окна File Open [Открыть файл]. Вы также можете выбрать другую иконку, нажав на текущую иконку (изображение кликабельно [spring-loaded]) — см. далее более подробную информацию. Нажмите OK. Обратите внимание, что ваш ярлык не появится на рабочем столе... пока. По сути, он добавляется в то же место, что и другие ярлыки — */usr/share/applications*, так что вам понадобится добавить его через Dash с помощью вышеописанной процедуры. Стоит проверить ваш новый ярлык рабочего стола и убедиться, что он работает, как вы и надеялись — в частности, если вы создали его с помощью редактора меню *Alacarte*, откройте терминал и напечатайте следующие команды:

```
cd Desktop  
desktop-file-validate <filename>.desktop
```

Если не выведется никаких ошибок, все должно быть в порядке.

## Редактирование ярлыков

Ярлыки легко перемещать, перетащив их в желаемое место. Вы можете быстро упорядочить их, нажав правой кнопкой мыши на пустом пространстве и выбрав Sort Desktop icons by Name [Сортировать ярлыки рабочего стола по имени]. По умолчанию иконки расположены в одну линию, чтобы обеспечить аккуратную равномерную сетку, но если вы предпочитаете некоторую хаотичность, просто щелкните правой кнопкой мыши по рабочему столу и уберите отметку с Align Desktop icons [Выравнивать ярлыки].

Вы также увидите появившуюся опцию Resize Icon [Изменить размер иконки] — выбрав ее, можно с помощью рукожоток, появившихся вокруг иконки, сделать ее больше или меньше, чтобы отдельные иконки легче привлекали ваше внимание. Используйте опцию Restore Icon Original Size [Вернуть к исходному размеру], чтобы быстро вернуть иконку к ее размеру по умолчанию. Растворимость иконки, конечно, в немалой мере зависит от ее максимального размера, но в большинстве случаев вы обнаружите, что вполне можете увеличить размер иконки вдвое, а то и вчетверо, без потерь детализации.

Если вам не по душе вид иконки по умолчанию, его тоже можно изменить; вы, конечно, можете отредактировать файл *.desktop* в текстовом редакторе вроде *laptop*, но куда удобнее будет просто щелкнуть правой кнопкой мыши по ярлыку и выбрать Properties, а затем по самой иконке. Откроется диалоговое окно, приглашающее вас выбрать другую иконку (в формате PNG или SVG). Вы можете выбрать готовую иконку — загляните в */usr/share/icons* на предмет разграбления существующих наборов иконок, или скачайте индивидуальные иконки с сайта наподобие [www.iconarchive.com](http://www.iconarchive.com); либо создайте свою собственную с нуля.

Чтобы создать свою собственную иконку, можно в *GIMP*, популярном приложении редактирования приложений, обрезать и изменить размер существующего изображения нужного размера — 192×192 должно быть достаточно для иконок рабочего стола при их максимальном уровне увеличения 400%. Убедитесь, что все сохранено в формате PNG. Или можно установить специальный инструмент для редактирования иконок — попробуйте установить *KIconEdit* из *Software Center*. Вы увидите, что он предлагает вам сеткообразный дисплей для создания иконок с нуля и поддерживает иконки размером до 200×200 пикселей. **LXF**



Хотите, чтобы на вашем рабочем столе отображались ключевые системные ярлыки, такие как ваша домашняя папка, сеть и корзина? Установите *Unity Tweak Tool* и выберите Desktop Icons под System. Просто щелкните по ярлыкам, которые вы хотите разместить на рабочем столе — и, о чудо, они появятся. Вы можете перемещать их, как и любые другие, перетаскивая и бросая.

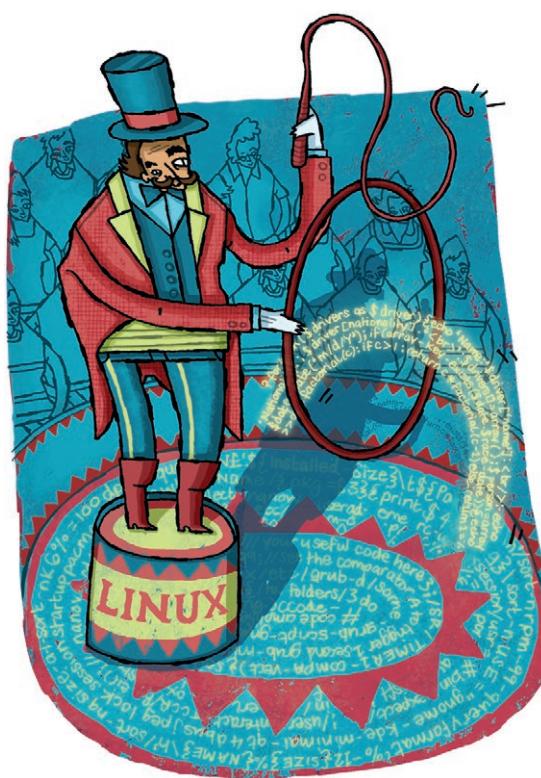
# Ubuntu: Чиним хромую загрузку

**Ник Пирс** знает, как решить проблемы с незагружающимся компьютером, не прибегая к кнопке «Паника» (или кувалде).



Наш эксперт

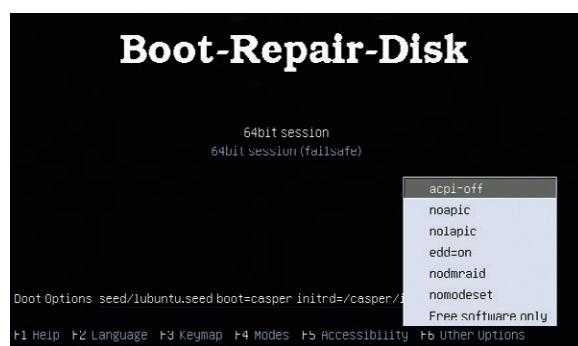
Ник Пирс провел уже столько жутких часов в борьбе с проблемами с загрузкой, и в таком количестве операционных систем, что все и не упомянуть.



**П**роблемы с загрузкой... Тот момент, когда — принимаясь за свою работу либо развлечения — вы вдруг видите загадочное сообщение об ошибке или, что еще хуже, пустой экран. И сколько бы вы ни перезагружали или ни сбрасывали компьютер, это непреодолимое препятствие стоит у вас на пути. Так что же делать?

Проблемы с загрузкой отличаются большим разнообразием, и их бывает трудно отследить. Однако существуют здравые принципы, позволяющие решить многие ошибки, и на этом уроке мы рассмотрим средства и методы, пригодные для устранения большинства проблем с загрузкой. Начните с того, как устроен процесс загрузки (см. врезку «Процесс загрузки», стр. 59). Вы узнаете, что этот процесс можно разделить на три больших этапа, связанных с загрузчиком *Grub 2*: до *Grub*, собственно *Grub* и после *Grub*. Это позволит сосредоточить свои усилия именно на том участке, где происходит ошибка или зависание процесса загрузки.

Начнем с начала. Вы включаете компьютер. Если питание поступает, но больше ничего не происходит, возможно, у вас есть какая-то аппаратная проблема. Если вы недавно копались во внутренностях своего компьютера, проверьте, что все подключено



Если диск Boot-Repair не загружается, позэкспериментируйте с вариантами безопасной загрузки.

правильно. Если нет, отключите все устройства, кроме клавиатуры, и попробуйте еще раз. Если это не помогло, аккуратно откройте корпус и отключите внутренние диски. Если компьютер стал загружаться до экрана заставки, можете снова подключить внутренние диски и попробовать еще раз; если теперь вы можете загрузиться до экрана входа в систему, выключите компьютер и начните подключать внешние устройства, чтобы посмотреть, сохранилась ли проблема и связана ли она с конкретным устройством. Если это так, попробуйте другой кабель или зайдите в Google известные проблемы, связанные с этим устройством.

Если вам повезет, то материнская плата издаст серию звуковых сигналов или миганий светофорами — и это пригодится для идентификации проблемы, опять же с помощью Интернета. Может потребоваться замена компонента или нечто посеребренное.

Если вы добираетесь до экрана заставки, но потом компьютер зависает или появляется сообщение «операционная система отсутствует», прежде всего припомните любые последние изменения. Например, если вы разгоняли процессор, войдите в EFI или BIOS и загрузите безопасные настройки по умолчанию. После этого снова попробуйте перезагрузиться.

Если не получится, то проблема скорее всего связана с жестким диском, и поэтому первым делом надо заглянуть в главную загрузочную запись и *Grub*.

Если меню *Grub* не появляется автоматически при включении компьютера, попробуйте перезагрузить его, удерживая Shift, или нажать Esc, чтобы открыть меню загрузки *Grub* и убедиться, что оно вообще может загрузиться. Если его нигде нет, перейдите к разделу про инструмент *Boot-Repair*.

Если *Grub* загружается, но вы не находите ни одной загружаемой ОС, есть несколько сценариев: может появиться базовая командная строка *grub>* или *grub rescue>*, это означает, что один или несколько файлов *Grub* отсутствуют или повреждены. Вы можете



Многие проблемы с *Grub* возникают при настройке двойной загрузки с Windows. Если ваша система не распознает Linux после его настройки, сверьтесь с нашей статьей о мультизагрузке (см. стр. 42). Там вы найдете помощь и советы по разрешению таких проблем.

## Процесс загрузки

По нажатию кнопки включения питания управление передается EFI или BIOS вашего компьютера, которые запускают различные компоненты, выполняют базовые диагностические проверки и пытаются найти загрузочное устройство, которым обычно является первый жесткий диск. После его обнаружения BIOS или EFI ищут главную загрузочную запись (Master Boot Record, MBR) в самом начале диска. В MBR содержится крошечная программа, которая загружает следующую часть загрузчика, считывая файл (например, **e2fs\_stage\_1\_5**), который, в свою очередь, может вызвать загрузчик *Grub*. Сообщение об «отсутствующей операционной системе» в этот момент означает, что чего-то не хватает — либо *Grub*, либо MBR, либо самого диска.

Успешно загрузившись, *Grub* считывает файл **menu.lst**, содержащий тот выбор, который вы видите в меню. Каждая запись

в **menu.lst** по сути определяет диск, раздел и файл, который содержит ядро Linux, а также файл RAM-диска, используемый ядром при загрузке. Запись также содержит любые дополнительные параметры, передаваемые ядру.

Затем управление передается ядру, которое пытается смонтировать корневую файловую систему. Это ключевой момент, и если попытка завершится неудачно, вы можете получить панику ядра или система может зависнуть. При успехе попытки система создает один процесс из файла **/sbin/upstart** (в других дистрибутивах используется **init**) — а если что-то пойдет не так, вы получите панику ядра, и система может снова зависнуть или перебросить вас в оболочку root. В этот момент *upstart* начинает запускать скрипты и события *upstart* для запуска других сервисов, и в конечном итоге на экране появляется окно входа в систему.

получить конкретное сообщение об ошибке или просто видеть только *Grub* и ничего другого — то есть загрузчик не может найти даже самой необходимой информации для продолжения загрузки.

Нажав **c**, вы сможете войти в режим терминала *Grub* и выполнить простые проверки или что-то исправить — например, запустить загрузку вручную, нажав **Ctrl+X** или **F10**, или выполнить набор команд для просмотра текущих настроек и изменений базовых настроек, например, графического режима. Полное руководство по инструментам для устранения неполадок в *Grub* доступно по ссылке <http://bit.ly/Grub2Troubleshooting>, но помните, что в большинстве случаев проще воспользоваться *Boot-Repair*.

Если меню *Grub* появляется и при выборе одного из пунктов меню система зависает, то проблема может быть связана с файлом настройки *Grub*. Но если Linux начинает загружаться и зависает только потом, то проблема связана с операционной системой; в этом случае перейдите к разделу «Неполадки вслед за *Grub*» (см. стр. 60).

## Инструмент Boot-Repair

Если вам не удается исправить проблемы с *Grub* вручную или если в системе вообще нет следов *Grub*, надо воспользоваться услугами диска аварийного восстановления и инструмента *Boot-Repair*, который работает со всеми дистрибутивами на базе Debian, включая Ubuntu. Инструмент *Boot-Repair* запустится автоматически при загрузке с диска *Boot-Repair*, а если вы не сможете создать этот диск,

но у вас есть установочный диск Linux, загрузите Live-среду, а затем и инструмент *Boot-Repair*, следующими командами:

```
$ sudo add-apt-repository ppa:yannubuntu/boot-repair
$ sudo apt-get update
$ sudo apt-get install boot-repair
$ boot-repair
```

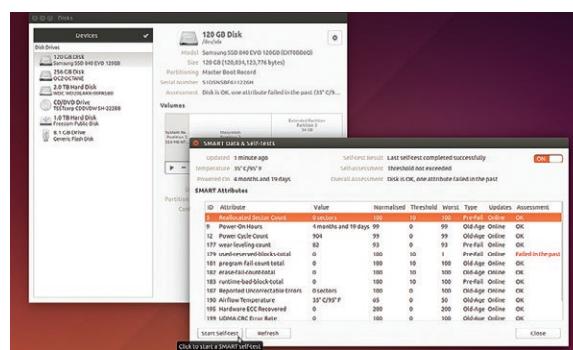
Инструмент *Boot-Repair* позволяет решить проблемы ранней стадии загрузки, вызванные загрузочным сектором жесткого диска, главной загрузочной записью и *Grub*. По сути это удобная и дружелюбная к пользователю графическая оболочка для утилит командной строки, требуемых для решения многих проблем. В *Boot-Repair* есть опция **Recommended repair** [Рекомендуемое восстановление], которая обещает решить большинство распространенных проблем. Или можно выбрать **Advanced options** [Дополнительные параметры], чтобы увидеть, на что способно средство, и выполнить необходимые исправления вручную, не пачкаясь с терминалом. В пошаговом руководстве (см. «Изменение настроек инструмента *Boot-Repair*», стр. 61) описаны возможные варианты восстановления и настройки, но учтите, что работа средства зависит от контекста, и многие параметры могут быть недоступны или отсутствовать, в зависимости от вашей системы.

Инструмент автоматически создает лог-файл действий вашей системы, который затем можно опубликовать на пользовательских форумах Ubuntu, чтобы спросить совета. Прежде чем предпринимать любые перенастройки, стоит сначала попробовать рекомендуемое восстановление, а затем попросить помощи на форуме с помощью лог-файлов — это подкрепит ваш выбор правильных действий и не ухудшит вашу ситуацию.

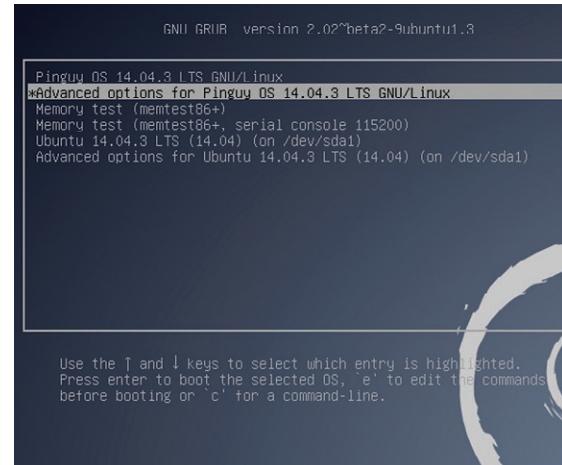
## Проблемы с загрузкой не из-за *Grub*

Если вы не дошли даже до загрузки *Grub*, пригодится диск аварийного восстановления. Загрузившись с диска, проверьте наличие и состояние вашего жесткого диска. Откройте файловый менеджер и убедитесь, что доступны разделы и файлы на них — тут самое время сделать копии всех ценных файлов, прежде чем продолжить.

Если ничего не появляется, проверьте, определяет ли система ваш жесткий диск, запустив утилиту *Disk* с Ubuntu Live CD — если вы пользуетесь диском *Boot-Repair*, то потребуется установить *gnome-disk-utility* с помощью менеджера пакетов *Synaptic*



▶ При загрузке с диска аварийного восстановления сразу же проверьте, работает ли ваш жесткий диск.



▶ Появление меню *Grub* — важный момент процесса загрузки. Если загрузка зашла так далеко, шансов восстановить систему гораздо больше.



При загрузке с диска аварийного восстановления компьютер автоматически определяет его и загружается с него. Если этого не произошло, откройте меню загрузки при запуске компьютера (обычно клавишей F11). Если не получилось, войдите в EFI или BIOS и выберите диск в качестве первого устройства загрузки.

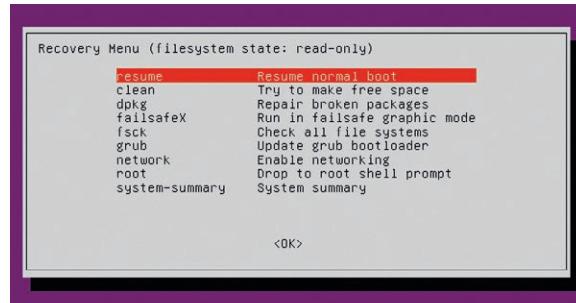
в разделе *System Tools* [Системные инструменты]. После установки откройте его из меню *Accessibility* [Доступность]. Утилита *Disk* выводит список всех физически подключенных дисков — если вашего нет, может оказаться, что он вышел из строя; тогда раскошевайтесь на новый диск и доставайте последнюю резервную копию, или начинайте с нуля со свежей установкой Ubuntu, заодно обзаведясь привычкой делать резервные копии системы. Если диск появился, выберите его из меню слева, где можно просмотреть его таблицу разделов и физическое состояние с помощью атрибутов SMART [Ред.: — Не такие уж они и умные.]. Не паникуйте, если диск не при последнем издохании, но обратите внимание на сами разделы. Если вы запустили средство *Boot-Repair*, его рекомендованные настройки включают полную проверку диска, но ее можно выполнить и вручную с помощью *GParted*, которая есть на обоих дисках аварийного восстановления.

*GParted* позволяет узнать, как организованы разделы, и определить, который из них загрузочный. Щелкните по нему правой кнопкой мыши и убедитесь, что пункт *Mount* [Монтировать] недоступен, затем выберите *Check to schedule a disk check using the fsck tool* [Запланировать проверку диска с помощью утилиты *fsck*]. После нажатия кнопки *Apply* [Применить] эта утилита проверит диск на наличие ошибок и попытается их исправить, но важно, чтобы диск не был смонтирован перед началом проверки. Обязательно дайте проверке завершиться — она может занять несколько часов или даже несколько дней, но ее отмена почти наверняка приведет к повреждению раздела. Убедитесь, что проверка выполняется на всех разделах загрузочного диска.

В большинстве случаев после выполнения этих проверок у вас по крайнем мере снова запустится *Grub*, если диск не поврежден физически или раздел не испорчен неправильно.

## Неполадки вслед за Grub

Если окажется, что *Grub* работает вроде нормально и проблемы начинаются при загрузке самого Linux, попробуйте во время загрузки переключиться в подробный режим, нажав клавишу *Esc*, чтобы посмотреть, нет ли каких-то зацепок в отладочных сообщениях (или — не виснет ли система в некий конкретный момент). Запишите эти сообщения и попробуйте скопировать их в свой любимый поисковик, чтобы найти какие-то советы в Интернете. Если не получилось, удерживайте *Shift* во время загрузки, чтобы открыть меню *Grub*, затем выберите *Advanced options* [Расширенные параметры] и '(recovery mode)' [режим восстановления]. При этом Ubuntu будет запущен в минимальном состоянии, а файловая система будет смонтирована в режиме только для чтения. Если это получилось, то прокрутив сообщения, вы должны увидеть меню восстановления с 9 пунктами.



➤ Режим восстановления Ubuntu позволяет проверить варианты решения проблемы, когда *Grub* работает, а Ubuntu — нет.

Эти пункты не требуют пояснений — например, *clean* [Очистка] пригодится, если жесткий диск переполнен, что и вызывает проблемы с загрузкой. Если проблемы начались из-за некорректной установки какого-то пакета, *dpkg* его подлечит, после чего система может вернуться в рабочее состояние.

Пункт *failsafeX* [Безопасный режим X] полезен, если при загрузке вы видите черный экран или не работает графический рабочий стол — фактически при выборе этого пункта обходятся проблемы с графическими драйверами или X-сервером, и вы получаете безопасный графический режим, в котором можете решать проблемы дальше.

Мы уже касались утилиты *fsck* — она проверит диск на наличие поврежденных файлов, что может избавить от многих ошибок, особенно если система «упала» и после этого перестала загружаться. Пункт *grub* не нужен, если вы не использовали собственные инструменты восстановления *Grub* вместо *Boot-Repair* при восстановлении загрузки — выбор этого пункта сделает изменения в *Grub* постоянными.

Выберите пункт *network* [Сеть], чтобы включить сеть, и пункт *root*, чтобы переключиться в оболочку и устранять неполадки оттуда. При этом обязательно смонтируйте файловую систему в режиме для чтения и записи командой *mount -o remount,rw /*.

При загрузке также можно передать Ubuntu временные параметры ядра, в некоторых ситуациях это помогает. Выбрав свою операционную систему в меню *Grub*, нажмите клавишу *e*, чтобы изменить файл ядра. Прокрутите вниз до строки, начинающейся с *linux* — параметры надо добавить в конец этой строки после *quiet splash*. Обязательно разделяйте параметры пробелами. Сделав это, нажмите *Ctrl+X* для загрузки с этими параметрами.

Учтите, что все добавляемые здесь параметры временные — то есть удаляются при следующей загрузке, поэтому можете экспериментировать с ними, пока не найдете рабочее решение,



Системные журналы — ценный источник информации для решения проблем. Их можно открыть из каталога */var/log* с помощью файлового менеджера вашего диска аварийного восстановления или редактора *laptop* в оболочке. В частности, обратите внимание на *syslog* и команду *obolochki dmesg*.

## Создание резервной копии на случай неудачи

Не считите за причуду, но при решении проблем с загрузкой постарайтесь прежде всего сделать резервную копию жесткого диска — она всегда позволит откатить систему до того состояния, в котором она находилась при первом возникновении проблемы. Конечно, если вы достаточно прилежны и регулярно делаете резервную копию системы, то просто сможете всегда откатить ее в рабочее состояние, но помните, что это может привести к потере данных, если ваш каталог *home* и установка Linux находятся на одном разделе (как по умолчанию и принято в Ubuntu).

Вам понадобится подходящее устройство для резервного копирования — обычно это

USB-диск — и средство создания образа всего системного диска. В Live-средах Ubuntu и *Boot-Repair* можно обойтись утилитой *dd*, но диск для резервного копирования по объему должен быть не меньше, а лучше — больше, чем диск, который вы копируете.

На другом конце диапазона сложности — инструмент *Redo Backup & Recovery*. Для записи его ISO-образа размером 261 МБ вам понадобится чистый CD или DVD, но зато вы получите удобный графический интерфейс. Образ можно создать с сайта [www.redobackup.org](http://www.redobackup.org) — руководство по его использованию см. в статье «Сбрасывать с Windows 10», стр. 26 *LXF207*.



➤ *Redo Backup & Recovery* предоставляет в общем простейший способ сделать безотказную копию всего жесткого диска.

а затем — при необходимости — сделайте изменения постоянными, изменив файл настройки *Grub* (`sudo nano /etc/default/grub`).

Параметры из Live-среды также можно передать, выбрав *Add a kernel option* [Добавить параметр ядра] в инструменте *Boot-Repair*. Список включает 15 распространенных параметров, способных помочь в устранении неполадок. Примеры включают `acpi=off`, который отключает систему ACPI, известную как причину случайных перезагрузок или зависаний системы на определенных компьютерах, и `nomodeset`, который велит Ubuntu загружать графические драйвера только после загрузки X, но не до нее. Эти временные параметры также можно передать и диску аварийного восстановления, если он не работает. Нажмите F6 на начальном экране загрузки, чтобы выбрать параметр. Чтобы подробнее узнать о параметрах, введите их в поисковую систему или зайдите на <http://tinyurl.com/KernelParametersList>, там есть полный список.

## Восстановление установки

И последнее, что можно сделать из загрузочного меню *Grub*. Если ваше ядро обновлялось, можно попробовать загрузить старую версию ядра в разделе *Grub Advanced options* [Расширенные параметры]. Вы увидите в списке все доступные версии ядра и сможете вернуть загрузку предыдущей версии, если считаете, что проблему

вызывало последнее ядро. Если это сработает, можно закрепить данную версию, отредактировав файл настройки *Grub* — проще всего это сделать через инструмент *Boot-Repair*.

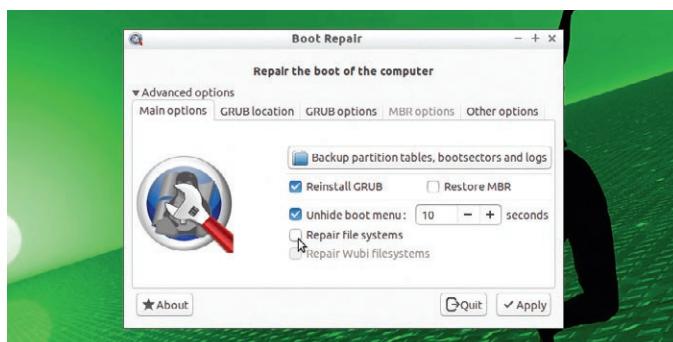
Если всё особенно безрадостно, можете попытать счастья, переустановив Ubuntu поверх себя. Загрузитесь с Ubuntu Live CD и в ответ на запрос выберите *Install Ubuntu* [Установить Ubuntu]. Когда появится окно *Installation type* [Тип установки], в нем возникнет новый вариант, по умолчанию уже выбранный — *Reinstall Ubuntu...* [Переустановить Ubuntu...].

В этом варианте по сути происходит переустановка Ubuntu, не затрагивающая каталог или раздел **home**; а значит, сохраняются не только ваши документы и другие файлы, но и ключевые настройки и многие программы. Также сохраняются пункты в меню загрузки, и вы не потеряете доступ к другим операционным системам.

При переустановке заменяются системные файлы, что должно искоренить все поврежденные файлы и оживить ваш компьютер. Хотя ваши пользовательские файлы затронуты не будут, советуем перед началом работы сделать резервную копию диска — или по крайней мере раздела или каталога **home**.

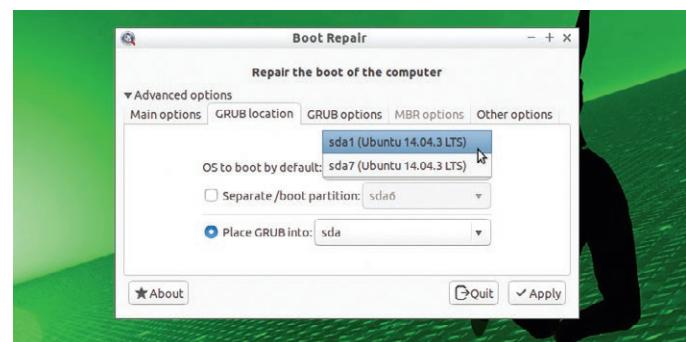
Для гарантии отсутствия потерь обязательно заново создайте во время установки все учетные записи пользователей с теми же логинами и паролями, включая, разумеется, вашу собственную. [LXF](#)

## Изменение настроек инструмента Boot-Repair



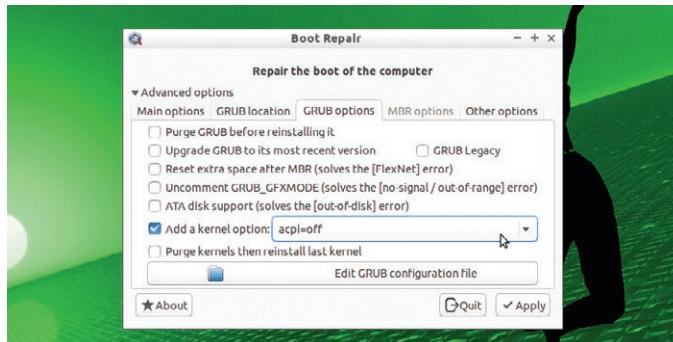
### 1 Основные параметры

На первой вкладке есть удобная кнопка для резервного копирования текущей таблицы разделов, загрузочного сектора и лог-файла — нажмите ее, чтобы скопировать эту важную информацию. Здесь также можно переустановить *Grub*, восстановить MBR и включить или отключить отображение меню *Grub*. Если вы думаете, что файловая система повреждена, поставьте галочку *Repair file systems* [Восстановить файловые системы], чтобы проверить ее и исправить ошибки.



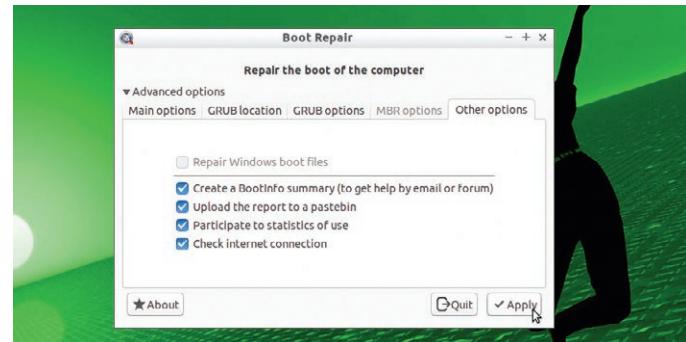
### 2 Расположение Grub

На этой вкладке вы можете выбрать, какую именно ОС загружать по умолчанию в мультизагрузочной конфигурации. Вы также можете разместить *Grub* на отдельном разделе `/boot` — обычно это бывает необходимо в случае зашифрованных дисков, дисков с LVM или некоторых старых компьютеров. Последний параметр указывает тот диск, на котором располагается сам загрузчик *Grub* (по умолчанию это `sda`).



### 3 Параметры Grub

Этот раздел начинается с параметров, гарантирующих обновление *Grub* до последней версии. Также предлагаются исправления трех специфических ошибок. Здесь также можно добавить новые параметры ядра в меню *Grub* или удалить все предыдущие ядра перед установкой последней версии. Также есть параметр, позволяющий редактировать файл настройки *Grub* напрямую.



### 4 Другие настройки

Если вкладка с параметрами MBR доступна, используйте ее, чтобы восстановить MBR из резервной копии и выбрать разделы, которые будут загружаться из нее. На последней вкладке, *Other options* [Другие параметры], можно восстановить файлы Windows (в большинстве случаев это не нужно) и скопировать информацию о ваших настройках в текстовом виде для удобного поиска в Интернете.

# Gentoo: Ставим через Portage

**Нейл Ботвик** выходит за рамки обычного руководства по обучению установке Gentoo, чтобы показать, как надо использовать гибкий дистрибутив.



Наш  
эксперт

Нейл Ботвик имеет большой опыт работы с загрузкой, ведь у него по компьютеру в каждой комнате, и гораздо меньший — с перезагрузкой, поскольку он перешел на Linux.



## Использование -uaN @world для установки пакетов с обновлениями или отмеченных флагом USE.

Время от времени мы пишем руководство для начинающих для начала работы с Gentoo — я это знаю, потому что как минимум одно написал сам. Они, как правило, основаны на первой части превосходной Gentoo Handbook по созданию рабочей системы. А потом пользователю остается читать инструкцию

и надеяться на себя, так что здесь мы рассмотрим основной элемент Portage и как его использовать для поддержания системы Gentoo после установки.

Название 'Portage' на самом деле обозначает два отдельных, но взаимосвязанных объекта. В первую очередь это сама система управления пакетами, используемая для установки, обновления и удаления пакетов. Кроме того, есть дерево портежей, т.е. коллекция сборочных файлов [ebuild], используемых Portage для управления установленным программным обеспечением. Это различие было не столь важно, но сейчас есть альтернативные менеджеры пакетов для Gentoo, который по-прежнему используют дерево портежей. Здесь мы рассмотрим только менеджер пакетов Portage, а при ссылке на коллекцию ebuild будем использовать термин «дерево портежей», во избежание путаницы.

В Gentoo почти все пакеты собираются из исходников. Педанты среди вас могут утверждать, что то же самое верно для пакетов Debian или Fedora, и будут правы (а кое-кого люди на вечеринках избегают). Разница в том, что в этих дистрибутивах разработчики компилируют программы и создают хранилища (репозитории) двоичных пакетов, эффективно архивированные; они распаковываются на жесткий диск (с некоторыми действиями по настройке). А в Gentoo компиляция выполняется пользователем при установке пакета. Вместо репозиториев двоичных пакетов дерево портежей в Gentoo содержит ebuild'ы, которые представляют собой короткие сценарии оболочки, описывающие, как загрузить исходный код, скомпилировать и установить его. В общей сложности это всего несколько сотен мегабайт, так что хранится локально. Оно синхронизируется с главным деревом, что нужно делать как минимум раз в пару недель, но не чаще раза в день (командой \$ emerge --sync). Если у вас есть последняя копия дерева портежей, вы, вероятно, захотите убедиться, что установленное программное обеспечение в актуальном состоянии: \$ emerge --update --deep --ask @world.

Имя пакета, которому предшествует @, означает набор, коллекцию пакетов. Есть два набора, определенные по умолчанию: @world содержит все программное обеспечение, которое вы установили, а @system — основные пакеты. Для этих двух наборов начальный @ не обязателен в силу исторических причин, так что можно встретить написание \$ emerge -uDa world. Значение --update должно быть очевидно; --ask перечисляет, что он будет делать, и ждет подтверждения (есть также --pretend), а --deep выполняет более глубокий анализ зависимостей (что увеличивает время на создание списка пакетов для установки и совершенственно не обязательно).

## Установка и удаление

А если вы что-то установили, но решили, что вам этого не надо? Можно использовать \$ emerge -Ca sys-apps/foo apps-misc/bar как

```
File Edit View Bookmarks Settings Help
These are the packages that would be merged, in order:
Calculating dependencies ... done!
[ebuild R ] sys-apps/coreutils-8.24 USE="acl* -hostname% -kill%"
[ebuild U ] sys-kernel/linux-headers-4.4 [4.3]
[ebuild U ] media-libs/exiftool-10.09 [10.07]
[ebuild R ] net-misc/rsync-3.1.2 USE="acl*"
[ebuild R ] sys-fs/ntfs3g-2015.3.14 USE="acl*"
[ebuild U ] dev-lang/swig-3.0.8 [3.0.7]
[ebuild R ] net-fs/cifs-utils-6.4 USE="acl*"
[ebuild R ] app-arch/libarchive-3.1.2-r2 USE="acl*"
[ebuild R ] sys-devel/gettext-0.19.7 USE="acl*"
[ebuild R ] sys-apps/sed-4.2.2 USE="acl*"
[ebuild R ] app-arch/tar-1.28-r1 USE="acl*"
[ebuild R ] app-backup/dar-2.4.17 USE="acl*"
[ebuild R ] app-cdr/cdrtools-3.02_alpha02 USE="acl*"
[ebuild U ] gnome-base/librsvg-2.40.13 [2.40.12]
[ebuild NS ] sys-kernel/gentoo-sources-4.4.0 [4.3.0, 4.3.2, 4.3.3] USE="-build exp"
[ebuild R ] app-editors/emacs-24.5-r1 USE="acl*"
[ebuild R ] sys-apps/systemd-228-r1 USE="acl*"
[ebuild R ] sys-apps/shadow-4.2.1-r2 USE="acl*"
[ebuild R ] net-print/cups-2.0.4 USE="acl*"
[ebuild R ] net-fs/samba-4.2.7 USE="acl*"
[ebuild R ] kde-frameworks/kio-5.17.0 USE="acl*"
[ebuild R ] kde-base/kdelibs-4.14.15-r1 USE="acl*"

would you like to merge these packages? [Yes/No] ■
```

```
# required by gnome-base/gvfs-1.26.2::gentoo
# required by xfce-base/thunar-1.6.10-r1::gentoo[udisks]
# required by xfce-base/xfdesktop-4.12.3::gentoo
# required by xfce-base/xfce4-meta-4.12::gentoo
# required by @temp
# required by @selected
# required by @world (argument)
>-dev-libs/libgdata-0.17.3 gnome
# required by gnome-base/gnome-keyring-3.18.3::gentoo
# required by app-crypt/libsecret-0.18.3::gentoo
# required by net-libs/gnome-online-accounts-3.18.2.1::gentoo
# required by dev-libs/libgdata-0.17.3::gentoo[gnome]
# required by gnome-base/gvfs-1.26.2::gentoo
# required by xfce-base/thunar-1.6.10-r1::gentoo[udisks]
# required by xfce-base/xfdesktop-4.12.3::gentoo
# required by xfce-base/xfce4-meta-4.12::gentoo
# required by @temp
# required by @selected
# required by @world (argument)
>-app-crypt/pinentry-0.9.6-r6 gnome-keyring
net-libs/liboauth -nss
-UU- :%%-F1 xfce      All L1   (Fundamental) -----
Note: file is write protected
```

» Файл в `/etc/portage/package.use` показывает изменения флага `USE` для определенных пакетов. Обратите внимание, что комментарии здесь разрешены и очень даже поощряются.

-С или `--unmerge`, что принудительно удаляет указанные пакеты. Хорошая идея взять в привычку с командой `emerge` использовать `-a` (`-ask`), это дает вам шанс вернуться, если вы сделаете ошибку. Это удаляет пакеты `foo` и `bar`, но при их установке *Portage* может также установить зависимости, например, `libfoo` и `libbar`. Надо ли их тоже удалять? Можете ли вы хотя бы вспомнить, какие они были? Не потребуются ли они для чего-то еще? Полные ответы дают команда `--depclean` или опция `-c`.

По команде без указания пакетов, типа `$ emerge -ca` ( обратите внимание на использование `-c` в противоположность `-C` выше), *Portage* просканирует систему на предмет любых пакетов, которые не являются зависимостями чего-нибудь в `@world`, и удалит их. Эта команда когда-то считалась потенциально рискованной, но уже не первый год она совершенно безопасна в использовании. Прежде чем продолжить, проверьте список пакетов. Как почти всегда в Gentoo: если вы хотите выстрелить себе в ногу, он будет счастлив предоставить заряженный пистолет.

Вместо `--unmerge` можно также использовать `--depclean` (теперь это рекомендуемый способ удаления пакета). Также можно использовать `$ emerge -caV sys-apps/foo`. Эта команда проверит, что `foo` не требуется для чего-то еще, прежде чем удалить его.

## Как прекрасен этот `@world`

Откуда *Portage* знает, что есть в `@world`? Откуда он знает, какие пакеты вам нужны, а какие там просто в качестве зависимостей? Ответ находится в файле `/var/lib/portage/world`. Это просто список установленных вами пакетов, так что если вы установили `sys-apps/foo`, *Portage* добавил `sys-libs/libfoo`, в файле `world` появится только `foo`. При удалении `foo` *Portage* знает, что если `libfoo` никому больше не требуется, `depclean` может удалить `libfoo`. Это приводит нас к классической ошибке новых пользователей, загрязняющих файл `world` установкой пакетов, которые должны быть там только в качестве зависимостей, чаще всего библиотек. Набор `@world` должен содержать только используемые программы, но не пакеты, которые они используют. Если вы просто хотите что-то наскоро попробовать, используйте опцию `--oneshot` (-1), чтобы не вносить это в `@world`, например: `$ emerge -1a libfoo`.

Флаги `USE` принадлежат к одним из самых полезных и самых непонятных функций *Portage*. При создании пакета они управляют **добавочными возможностями**: например, если добавить сканер в `USE` при установке `Wine`, он будет построен с поддержкой использования сканера в программах `Windows`. Включение `-scanner` в `USE` при установке `Xsane` не будет иметь смысла, поскольку `Xsane` и сама требует сканер. Флаги `USE` могут быть заданы тремя способами. Для глобального использования они устанавливаются

## Атомные числа

По всей документации Gentoo вы увидите ссылки на атомы. Атом — это спецификация пакета, как `'gimp'` или `'media-gfx/gimp'`. При установке файлов часть с категорией является обязательной, лишь если есть пакеты с одинаковыми названиями в разных категориях. В атоме можно также указать версии, что может быть полезно, например, в `package.use`:

```
sys-apps/foo
= sys-apps/foo-1.2.3
~ sys-apps/foo-1.2.3
< sys-apps/foo-1.2.4
```

Первое соответствует любой версии `foo`, второе — только версии 1.2.3, третье соответствует версии плюс любые незначительные пересмотры, как `foo-1.2.3-r1`, а последнее соответствует любой версии старше 1.2.4.

в `/etc/portage/make.conf`, который вы создали во время установки. Для отдельного пакета их можно задать в командной строке:

```
$ USE="foo" emerge -1a bar
```

Хотя это полезно для тестирования, такой параметр будет переопределён при следующем появлении данного пакета. *Portage* обеспечивает механизм изменения флагов `USE` для отдельных пакетов путем перечисления их в `/etc/portage/package.use`. Это может быть файл или каталог. Если это каталог, учитываются содержимое всех файлов в нем. По мере добавления исключений вы можете обнаружить, что это упрощает организацию. Каждая строка представляет собой атом пакета и список флагов `USE`, которые к нему применяются, например, `sys-app.foo bar -baz`.

Команда `emerge` имеет пару ключей для действий с изменениями флагов `USE`, либо редактированием этих файлов, либо изменениями в `ebuild`:

```
$ emerge -a --newuse @world
$ emerge -a --changed-use @world
```

С `--newuse` (или `-N`) все пакеты, для которых изменились флаги `USE`, пересобираются. Второй вариант только переустанавливает пакеты, у которых используются измененные флаги, что позволяет избежать ненужных пересборок. Когда вы `emerge` какой-либо пакет с `-a` или `-p`, *Portage* показывает ваши флаги `USE` для пакета и их состояние, но не говорит вам, что они означают. Установив `app-portage/euses`, вы можете увидеть описание каждого флага `USE`:

```
$ euses scanner
scanner - Add support for scanner hardware (e.g. build the sane
frontend in kdegraphics)
net-print/hplip:scanner - Enable scanner on multifunction devices
which support it
```

Если вы думаете о смене флага `USE` глобально и хотите знать, какие из пакетов будут затронуты, `eqquery hasuse scanner` покажет вам все установленные пакеты, учитывающие этот флаг. *Eqquery* является частью `app-portage/gentoolkit`, который стоит установить сразу.

Есть и другие пакеты, которые стоит установить:

» **app-portage/eix** Создает базу данных для поиска доступных и установленных пакетов; также включает программу `eix-sync`, которая обновляет дерево портежей и базу данных одной командой.

» **app-portage/genlop** Работает с файлами журнала `emerge` и может дать полезную информацию, включая прогнозы продолжительности текущей операции.

» **app-portage/portage-utils** Содержит утилиты Q: `qlist` показывает файлы, установленные пакетом; `qfile` расскажет, какой пакет установил конкретный файл; а `qlop` выполняет функцию, аналогичную `genlop`.

Много чего еще есть в *Portage*; здесь мы рассмотрели некоторые основы, в следующем месяце охватим больше. [LXF](#)

# Procenv: Тест окружения

**Джеймс Хант**, создатель *procenv*, описывает свою мощную утилиту, показывающую все возможные аспекты среды, в которой она работает.



Наш эксперт

Джеймс Хант является разработчиком СПО и трудится в Intel Open Source Technology Center над дистрибутивом Clear Linux.

**Н**а этом уроке мы рассмотрим инструмент командной строки *procenv*. Он был написан для решения конкретных реальных проблем, но полезен во многих ситуациях. Впервые, лирическое отступление. Представьте себе следующий сценарий: 7 вечера, и у программиста-аса Алисы возникла проблема — она исправила критическую ошибку клиента, но отдел испытаний сообщил, что теперь приложение падает при запуске на тестовом сервере.

Алиса удивляется, как это возможно — ведь обе системы используют одну версию ОС, а в ее системе приложение отлично работает. Если она немедля не выяснит, чем ее система отличается от тестовой, ее ждет долгая ночь в офисе. Недолго думая, она пишет сценарий оболочки, чтобы попытаться выявить различия между двумя системами:

```
#!/bin/bash
echo -n "user: "; whoami
echo -n "directory: "; pwd
echo "environment variables: "; env
echo "ulimit: "; ulimit -a
echo "file descriptors: "; ls -l /proc/self/fd/
```

Для тех из вас, кому еще не приходилось писать такой скрипт (или кто просто отказывается это признать), скрипт этот предназначен для запуска в двух местах:

- » В среде, где ваша программа работает (возможно, на вашем рабочем компьютере или в системе разработки).
- » В среде, где ваша программа упорно отказывается работать (скорее всего, в тестовой или производственной системе).

Алиса надеется: сравнив вывод скрипта в обеих системах, она сумеет выявить различия, которые дадут ключ к пониманию, почему рабочее приложение падает. К сожалению, ночь у Алисы будет длинной, потому что скрипт просто не дает достаточно подробностей. Эх, знай она только про инструмент, который обеспечивает идеальное решение: *procenv*...

(Оговорка: Джеймс Хант написал *procenv*, чтобы решить пару интересных задач — см. врезку об истории *procenv* на с. 65) Это очень простой инструмент, который по умолчанию выводит все возможные аспекты среды, в которой он работает. Термин «среда» нуждается в некотором уточнении; *procenv* использует термин «среда» в довольно либеральном смысле, стирая границы между «процесс» и «система». По сути, он запрашивает все возможные аспекты среды, в которой работает, с точки зрения процесса.

Если у вас есть некоторый опыт программирования на языке оболочки, вы можете полагать, что процесс среды — это просто набор пар переменных имя/значение, которыми можно управлять с помощью консоли:

```
$ var=foo
$ echo $var
```

```
procenv --help | less
--output=<type> : Send output to alternative location.
                   <type> can be one of:
                     file   : Send output to a file.
                     stderr : Write to standard error.
                     stdout : Write to standard output (default).
                     syslog : Write to the system log file.
                     terminal : Write to terminal.

-p, --process    : Display process details.
-P, --platform  : Display platform details.
-q, --time       : Display time details.
-r, --ranges     : Display range of data types.
--separator=<str> : Specify string <str> as alternate delimiter
                   for output (default=:').
-s, --signals    : Display signal details.
-S, --shared-memory : Display shared memory details.
-t, --tty        : Display terminal details.
-T, --threads   : Display thread details.
-u, --stat       : Display stat details.
-U, --rusage     : Display usage details.
-v, --version    : Display version details.
-w, --capabilities : Display capability details (Linux only).
:
```

У *procenv* больше опций, чем у агента по недвижимости!

```
foo
Ладно, теперь спросим procenv:
$ procenv --environment | wc -l
105
```

Здесь я попросил *procenv* ограничить отображаемый вывод только переменными среды и посчитал строки вывода командой *wc*. Теперь давайте снова запустим *procenv*, но на сей раз без ограничений, чтобы вывести об окружении всё, что ни есть:

```
$ procenv | wc -l
4238
```

Итак, вы видите: окружение — отнюдь не одни переменные. На самом деле, *procenv* выдает настолько большой вывод, что полный дамп здесь показывать непрактично.

## Изобилие PID

После переменных окружения, следующим наиболее очевидным атрибутом среды является его идентификатор процесса или PID. При запуске *ps* или *top* можно увидеть, что каждый процесс имеет уникальный номер:

```
$ echo "current PID=$$"
current PID=1234
```

Если вы создаете новый процесс, запуская *Bash* еще раз (чтобы создать подоболочку), вы получите новый PID, соответствующий новой оболочке:

```
$ bash -c 'echo "sub-shell PID=$$"
sub-shell PID=1235
```

Чтобы подогреть аппетит, давайте рассмотрим, как *procenv* показывает это, запустив его дважды в Ubuntu и сравнивая результаты, в одном затейливом однострочнике:

```
$ diff -u <(procenv --process) <(procenv --process)
--- /proc/self/fd/11 2015-10-14 20:17:03.147630023 +0100
```



Если вы предпочитаете более дружелюбный инструмент, чем спартанский *diff(1)*, есть популярные графические интерфейсы к *diff* — *meld* и *kdiff3*. Кому нужны отличные характеристики, поддержка очень больших файлов изменений и невероятная скорость, тем следует рассмотреть *xxdiff*.



```
+++ /proc/self/fd/12 2015-10-14 20:17:03.147630023 +0100
@@ -1,10 +1,10 @@
process:
- process id (pid): 23893
+ process id (pid): 23894
parent process id (ppid): 23320
session id (sid): 23320 (leader=no)
name: 'procenv'
- ancestry: 23893 ('procenv'), 23320 ('zsh'), 23308 ('gnome-terminal-'), 16776 ('upstart'), 16660 ('lightdm'), 16597('lightdm'), 1 ('systemd')
- process group id: 23893 (leader=yes)
+ ancestry: 23894 ('procenv'), 23320 ('zsh'), 23308 ('gnome-terminal-'), 16776 ('upstart'), 16660 ('lightdm'), 16597 ('lightdm'), 1 ('systemd')
+ process group id: 23894 (leader=yes)
foreground process group: 23895
terminal: '/dev/tt1'
has controlling terminal: yes
```

Здесь мы видим довольно много PID. В этом примере выполняется два экземпляра `procenv --process`, которые велят `procenv` показать только детали процесса. Как можно видеть, PID изменился с 23893 на 23894.

Поскольку команды `procenv` в команде `diff` заключены в скобки, они будут выполняться в отдельных подоболочках, а поскольку символ `<` указан перед подоболочкой, вывод каждой подоболочки будет передаваться в `diff` (никаких временных файлов не требуется). Мы используем формат 'unified diff' (`-u`), поскольку его легче читать (плюс `+` в первой колонке обозначает добавление, а минус — удаление).

Записи родословной весьма интересны: они демонстрируют PID `procenv`, его родительский PID и PID'ы всех других «предков» вплоть до PID 1, демона init. Они, по существу, прослеживают одну из «ветвей» всех PID в системе.

Наконец, любопытная особенность этой родословной заключается в том, что она показывает и *Upstart*, и *Systemd*. Это потому, что *Systemd* теперь используется в Ubuntu как демон init, а *Upstart* до сих пор используется для управления графическим сеансом пользователя.

Ну, а что имеется в окружении, помимо переменных и PID? Да немало — фактически так много, что `procenv` должна группировать вывод по категориям: последняя версия `procenv` (на момент

написания, 0.42) отображает информацию в 36 разных категориях. Исчерпывающий перечень всего, что показывает `procenv`, давать не слишком практично, но имена параметров командной строки дают подсказку о том, что они умеют отображать: например, `--arguments`, `--compiler` и `--libc`. В качестве примера, `--libc` показывает детали из стандартной библиотеки. Для подробной информации см. `procenv --help` или тап-страницу (`procenv(1)`).

Давайте начнем с простого примера. Примечание: функция `--file=` полезна для записи вывода `procenv` без возможного иска-жения этих результатов из-за использования перенаправления консоли. Сначала запустите `procenv` как обычный пользователь и запишите вывод командой `$ procenv file=/tmp/procenv.log`. Затем запустите `procenv` как другой пользователь (здесь — пользователь `foo`) и запишите вывод командой `$ sudo -u foo procenv --file=/tmp/procenv-foo.log`. И, наконец, сравнимте результаты командой `$ diff /tmp/procenv.log /tmp/procenv-foo.log`.

Если вы собираетесь использовать формат вывода JSON, рекомендуется установить отличную утилиту `jq`, которая позволяет с легкостью извлекать элементы. Все переменные `procenv` описаны на его тап-странице.

## Расписание процессора

Большинство современных систем имеют более одного (виртуального или физического) процессора. Проведите такой эксперимент: `$ diff <(procenv --cpu) <(procenv --cpu)`. В этом примере выполняются два экземпляра `procenv --cpu`. Поскольку команды заключены в скобки, они будут выполняться в отдельных подоболочках, а поскольку символ `<` указан перед подоболочкой, вывод каждой подоболочки будет подаваться в `diff`. Вопрос на засыпку: почему команда выше иногда не дает вывод?

Алиса могла бы использовать это для сравнения двух систем, запустив `procenv` на обеих и сравнив результаты:

```
# На машине Алисы
$ procenv --file=/tmp/procenv-alice-workstation.log
# На тестовой системе
$ procenv --file=/tmp/procenv-test-env.log
```

А что еще можно сделать? Вот несколько идей. Некоторые результаты могут вас удивить. Не выяснить ли, что на самом деле делает `nohup cmd &`, запустив `procenv` обычным образом и записав вывод:

```
$ procenv --file=/tmp/procenv.log
```

Назовем это «базовым» выводом. Теперь запустим его с `nohup`, отправив в фон с помощью `&` и сравним два полученных файла:

```
$ nohup procenv --file=/tmp/procenv-nohup.log &
```

Сравним его с

```
$ procenv --file=/tmp/procenv-foreground.log &
```

»

## Хантова история `procenv`

Идею `procenv` породили два желания, которые грызли меня постоянно, и оба касаются системы инициализации *Upstart*, ранее используемой в Ubuntu.

Во-первых, мне был нужен подтверждающий инструмент для сценарного тестирования. *Upstart* имеет отличные модульные тесты, но я хотел создать набор «заданий» *Upstart* для запуска на реальной системе. Эти задания будут подтверждать каждый атрибут своего окружения, ожидаемый при выполнении (окружение работы *Upstart* должно соответствовать очень строгим определениям, как описано в тап-странице `upstart(5)`). Моя первоначальной мыслью было написать инструмент, способный выдавать сообщение;

если утверждение ложно, то инструмент выводит ошибку и завершается. Однако мне не нравилась идея передачи такого утверждения инструменту. Более ясная конструкция должна иметь инструмент, просто отображающий полное окружение, позволяя заданию проверить вывод инструмента с помощью `grep(1)`.

Вторым желанием было найти способ отладки удаленного тестирования сбоев. Тесты *Upstart* имеют решающее значение для обеспечения ожидаемого поведения системы и защиты от регрессий: если PID 1 не выполняется, система падает! Однако тесты, которые отлично делали свое дело в системе разработки, иногда по загадочным причинам проваливаются в разных других системах.

Я обнаружил, что решение этих проблем крайне сложно, поскольку системы защищены паролями. Это означало, что отладка зачастую ограничивается чтением журналов сборки!

Поскольку я не мог войти, следующим вариантом было точно узнать, чем рабочая среда отличается от систем разработки.

Создание инструмента `procenv` решило обе эти проблемы, поскольку он не только показывает всё окружение при запуске, но также запускается сам, когда собирается на удаленных серверах, которые я упомянул. Это предоставляет простую отладку путем сравнения журналов сборки `procenv` (содержащих весь вывод `procenv`) с выводом `procenv`, работающего на локальной системе.

» Подпишитесь на печатную или электронную версию на [www.linuxformat.ru/subscribe](http://www.linuxformat.ru/subscribe)!



Если вы хотите понять, что именно `procenv` делает, запустите его через `strace -f $ strace -o /tmp/strace.log`. На main странице `procenv` перечислены тап-страницы консультаций по каждой функции, чтобы узнать больше: например, раздел `--clocks` направит вас на тап-страницу `clock_gettime(2)`.

## » Procenv раскрывает детали сетевого интерфейса.

```
$ $$SHELL -c 'procenv --file=/tmp/procenv-background-subshell.log &'
```

Можно узнать, что делает перенаправление оболочки, сравнив с базовым выводом следующее:

```
$ procenv --file=/tmp/procenv-redirect.log </dev/null>/dev/null
```

Чтобы определить, что делает `script(1)`, сравните с базовым выводом следующее:

```
$ script -c 'procenv --file=/tmp/procenv-script.log'
```

Мы также можем определить, отличается ли запуск команды в графической среде от запуска ее в консоли, запустив `procenv --file=/tmp/procenv-emulator.log` в `gnome-terminal` (или в вашем любимом эмуляторе терминала). Затем запускаем `procenv` в настоящем терминале с помощью `Ctrl+Alt+F1`, входим и запускаем `procenv --file=/tmp/procenv-console.log`.

Можно также попробовать выполнить сравнение различных оболочек, скомандовав

```
$ sh -c 'procenv --file=/tmp/procenv-sh.log'  
$ bash -c 'procenv --file=/tmp/procenv-bash.log'  
$ zsh -c 'procenv --file=/tmp/procenv-zsh.log'
```

## Среда инициализации системы

Очень часто разработчикам и системным администраторам требуется полное понимание, какую среду обеспечивает инициализация системы. Вот как можно это сделать для систем на базе `Upstart`: во-первых, создайте файл `/etc/init/procenv.conf`:

```
description "Display Upstart process environment"  
task  
exec procenv --file=/tmp/procenv-main.log  
pre-start exec procenv --file=/tmp/procenv-pre-start.log  
post-start exec procenv --file=/tmp/procenv-post-start.log  
post-stop exec procenv --file=/tmp/procenv-post-stop.log
```

Далее запустите задачу `$ sudo start procenv`. После этого проверьте лог-файлы в `/tmp`.

Альтернативно, в дистрибутиве Ubuntu можно выполнить тот же тест, создав файл `$HOME/.config/upstart/procenv.conf` и сгенерировав журнал, запустив (без sudo) `$ start procenv`. Если вы используете системы на базе `Systemd`, вам потребуется создать файл `/lib/systemd/system/procenv.service`, содержащий

```
[Unit]  
Description=Display systemd process environment  
[Service]  
Type=oneshot  
ExecStart=/usr/bin/procenv --file=/tmp/procenv-main.log
```

И сгенерировать журнал:

```
$ sudo systemctl enable procenv.service
```

```
$ sudo systemctl start procenv.service
```

Проверить лог-файлы можно опять-таки в `/tmp`.

Иногда вы желаете знать о среде то, что изначально кажется невозможным наблюдать. Фокус в том, чтобы использовать функцию `procenv --exec`, использующую один из системных вызовов 'exec', который заменит текущий процесс другим.

Представьте, что вам нужно увидеть точное окружение, в котором запущена конкретная программа. Возможно, что эта программа вызывается другой программой в сложной цепи скриптов и других программ. Как это можно сделать? Да легко! Представьте, что мы хотим видеть среду, в которой вызывается `awk`. Во-первых, мы перемещаем реальный бинарник `awk`:

```
$ sudo mv /usr/bin/awk /usr/bin/awk.real
```

Затем создаем скрипт — под названием, допустим, `procenv.sh` — содержащий

```
#!/bin/sh  
cmd=$(basename "$0")  
exec /usr/bin/procenv --file=/tmp/procenv-$cmd.log --exec  
-- /usr/bin/$cmd.real "$@"
```

Далее мы устанавливаем программу как `/usr/bin/procenv.sh` командой `$ sudo install procenv.sh /usr/bin` и создаем символическую ссылку из `/usr/bin/awk` на `procenv.sh` командой `$ sudo ln -s /usr/bin/procenv.sh /usr/bin/awk`.

После этого фокуса, всякий раз, когда выполняется символическая ссылка `/usr/bin/awk`, на самом-то деле запускается скрипт `/usr/bin/procenv.sh`, определяющий имя команды, которым она называлась (в данном случае, `awk`). Затем скрипт перезапустится, запустив `procenv` с тем же PID, что и у скрипта, а `procenv` запишет свой вывод, в данном случае в `/tmp/procenv-awk.log`. После этого `procenv` лично перезапустит указанную команду, в данном случае — `/usr/bin/awk.real`. Важно подчеркнуть, что всё указанное будет работать с одним и тем же PID: `procenv.sh`, `/usr/bin/procenv` и `/usr/bin/awk.real`.

## Окружение PID 1

Что, если бы вы хотели узнать об окружении, создаваемого ядром для демона `init`, который запускается с PID 1? Поскольку демон `init` должен быть первым процессом, запущенным в системе (и, таким образом, получить себе PID 1), уже по его определению `procenv` нельзя запустить раньше демона `init`, верно? Неверно.

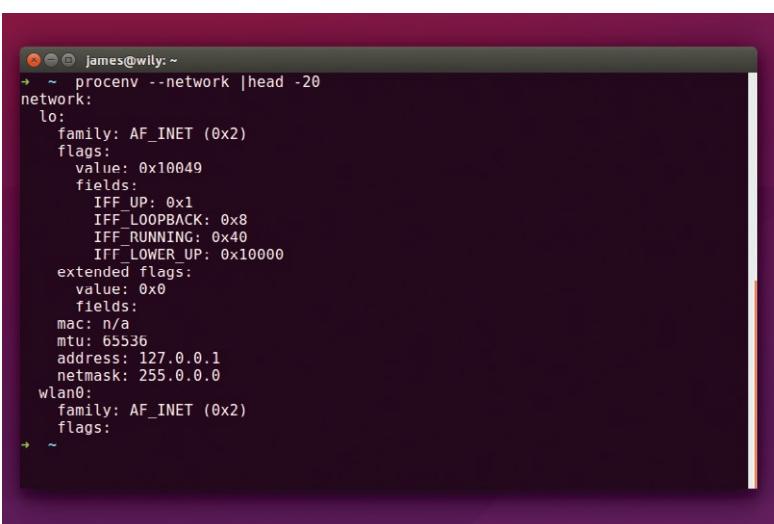
Можно снова использовать трюк с 'exec', поскольку это позволяет `procenv` работать как PID 1 и затем, закончив работу, передать тот же «идентификатор» PID 1 демону `init`. Синтаксис требует выполнять этот трюк несколько более запутанно, чем в примере выше. Это обусловлено тем, что у ядра нет универсального механизма передачи параметров демону `init`. Следовательно, мы заставим ядро задать две волшебные переменные для `procenv`, чтобы сказать ему, куда писать вывод и кому передать команду по завершении:

```
init=/usr/bin/procenv PROCENV_FILE=/dev/ttys0PROCENV_EXEC="/lib/systemd/systemd"
```

Учитите, что вам требуется ядро, способное преобразовывать «неизвестные» параметры командной строки ядра в переменные среды. Ubuntu делает это, но ядра других дистрибутивов, возможно, не могут. Полный список переменных, поддерживаемых `procenv`, см. на тап-странице `procenv(1)`.

## Форматы вывода

Со временем `procenv` получил несколько разных форматов вывода. В настоящее время он поддерживает: структурированный простой



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

текст (по умолчанию), XML, JSON и сиimb. Формат вывода по умолчанию проще всего для восприятия человеком. XML и JSON делают вывод *procself* бесконечно более подходящим для других инструментов. Загадочный сиimb требует небольшого пояснения. Это сокращение от «хлебных крошек», и он предназначен для удобного парсинга: данные отображаются в плоском формате, где каждое значение — в отдельной строке, а перед ним — все необходимые заголовки, через текущий разделитель. Вероятно, здесь поможет пример. Поскольку весь вывод *procself* использует внутри тот же набор процедур, даже флаг *--version* может давать вывод в любом из поддерживаемых форматов (на рис. справа).

Как можно видеть, в выводе по умолчанию отображается блок одной «версии». Этот блок содержит три значения: название, версию и автора. В формате сиimb отношение родитель/потомок является в некотором смысле более явным, поскольку каждая строка начинается с версии: т. е. все эти строки отображают информацию из блока «версии». Это может показаться очевидным, но если вы запустите *procenv --format=crumb*, вы увидите, что есть много разных префиксов блоков и помимо версии. Можно слегка улучшить вывод, добавив опцию *--separator*:

```
$ procenv --format=crumb --separator=: --version
version:name:procenv
version:version:0.42
version:author:James Hunt <jamesodhunt@ubuntu.com>
```

Заметьте, что раздражающий пробел перед значением теперь исчез. Можно улучшить еще немного:

```
$ procenv --format=crumb --crumb-separator='-' --separator='-
--limits
limits,RLIMIT_AS,soft,current,18446744073709551615
limits,RLIMIT_AS,soft,max,18446744073709551615
limits,RLIMIT_AS,hard,current,18446744073709551615
limits,RLIMIT_AS,hard,max,18446744073709551615
```

Я привел только несколько первых строк вывода, но теперь у нас есть *procself*, чтобы перечислить несколько ограничений ресурсов в формате CSV (значения через запятую). Это означает, что можно взять вывод и загрузить его прямо в приложение электронных таблиц.

И последний пример — версия 0.42 *procenv* в полной мере может воспользоваться преимуществами UTF-8. Это означает, что вы можете делать сумасшедшие вещи (см. ниже), чтобы показать детали всех существующих тактов в вашей системе, не без «шика» указав пользовательские разделители в юникоде:

```
$ procenv --format=crumb --separator=">" --crumb-separator="
>> " --clock
clocks >> CLOCK_REALTIME >> resolution -> 0.000000001s
clocks >> CLOCK_MONOTONIC >> resolution -> 0.000000001s
clocks >> CLOCK_MONOTONIC_RAW >> resolution ->
0.000000001s
clocks >> CLOCK_PROCESS_CPUTIME_ID >> resolution ->
0.000000001s
clocks >> CLOCK_THREAD_CPUTIME_ID >> resolution ->
0.000000001s
```

## Местоположение вывода

Иногда писать в стандартный вывод недостаточно, поскольку *stdout* может быть перенаправлен, например, в */dev/null*, вот почему *procself* поддерживает различные места для вывода: стандартный вывод (по умолчанию), стандартная ошибка, файл, системный журнал (управляется *Syslogd* или *journald* из *Systemd*) и прямо в терминал. Обратите внимание, что перенаправления консоли не требуется — просто переключите расположение записи вывода.

Если у вас установлена Ubuntu 14.04 (Trusty) или более новая (или последний Debian), установите *procself* обычным способом: *\$ sudo apt-get install procenv*. Инструмент распространяется под лицензией GPLv3, а исходный код находится в свободном доступе

```
james@wily: ~
$ procenv --version
version:
  name: procenv
  version: 0.43
  author: James Hunt <jamesodhunt@ubuntu.com>
$ procenv --format=json --version
{
  "version" : {
    "name" : "procenv",
    "version" : "0.43",
    "author" : "James Hunt <jamesodhunt@ubuntu.com>"
  }
}
$ procenv --format=xml --version
<?xml version="1.0" encoding="UTF-8"?>
<procenv version="0.43" package_string="procenv 0.43" mode="non-privileged" format_version="12">
  <section name="version">
    <entry name="name">procenv</entry>
    <entry name="version">0.43</entry>
    <entry name="author">James Hunt &lt;jamesodhunt@ubuntu.com&gt;</entry>
  </section>
</procenv>
$ ~
```

на GitHub (<https://github.com/jamesodhunt/procenv>). Если у вас возникнут проблемы или есть идеи по его улучшению, прошу завести ошибку на GitHub. Посмотрите, доступен ли *procself* для вашей системы. Если нет, то пакеты для большинства дистрибутивов есть на <http://bit.ly/ProcenvPackages>. А если вы хотите собрать его из исходников, следуйте обычному простому шаблону:

```
$ git clone https://github.com/jamesodhunt/procenv
$ cd procenv
$ ./configure && make && sudo make install
```

## Реальные данные

Хотите узнать, на что похожа среда сборки Travis-CI? Или среды сборки, используемые для сборки пакетов Debian или Ubuntu? Просто зайдите в раздел Results на странице GitHub, где есть ссылки на все публично доступные журналы сборки *procself*.

Так что, если вы пытаетесь отладить проблему, из-за которой ваше приложение не выполняется в, например, среде Travis-CI, просто обратитесь на <https://travis-ci.org/jamesodhunt/procenv>, посмотрите логи и сравните их с выводом *procself* при запуске в «рабочей» среде.

Чем больше платформ, на которых доступен *procself*, тем он полезней. В настоящее время он работает на Linux, FreeBSD, GNU Hurd и Android. Если вы можете помочь портировать его на другие системы (например, AIX, HP-UX, Solaris, Minix или OpenVMS), пожалуйста, свяжитесь с автором. На данный момент должно быть ясно, что окружение процесса содержит многое больше, чем просто переменные окружения. Вы также видели, как легко запустить *procself* и исследовать скрытые области процесса, и сколько он может показать о вашей среде.

Отличный, хотя и лаконичный обзор процессов и групп можно увидеть на ман-странице *credentials(7)*. Это одна из тех жемчужин, которые широко не рекламируются, и без содействия найти их непросто. Ман-страница *procself* содержит указатели на все команды, похожие на него. Эти команды по отдельности способны предоставить часть информации, предоставляемой *procself*. Ман-страница также дает ссылки на все необходимые системные и библиотечные вызовы для каждой опции, если вы хотите узнать больше о категории.

Если вы предпочитаете менее лаконичное объяснение, чем скучные ман-страницы, взгляните на следующие две книги: *Advanced Programming in the Unix Environment* (2-е издание) У. Ричарда Стивенса и Стивена А. Паго [W. Richard Stevens, Stephen A. Pago] и *Advanced Unix Programming* (2-е издание) Марка Дж. Рочкинда [Marc J. Rochkind]. Не пугайтесь слова «продвинутый» в названиях — обе вполне понятны и изложены в медленном темпе, они исчерпывающие и изобилуют небольшими примерами, закрепляющими пройденный материал. [LXF](#)

Каждая опция способна давать вывод в различные форматы.

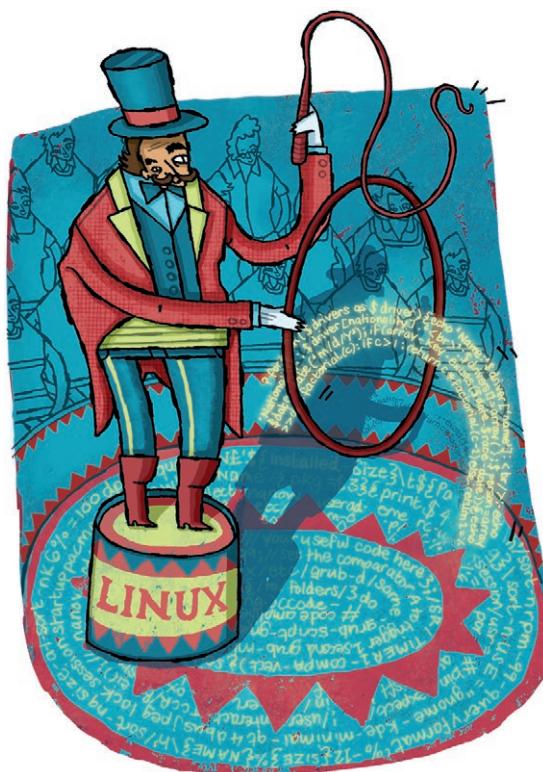
# Hadoop: Для Больших Данных

Данные — это валюта будущего. Рафик Рехман представляет навыки обработки и анализа данных, на которые есть спрос.



Наш  
эксперт

Рафик Рехман — автор и консультант, интересующийся новыми технологиями, обработкой и анализом данных, IoT и защитой информации.



Из-за огромного количества наборов данных, порожденных электронной коммерцией, производственными процессами, историей посещения сайтов, IoT и других источниками, все разновидности бизнеса ищут способы капитализировать эти данные. Эта разработка создала огромную потребность в наборе навыков обработки и анализа Большых Данных, и поэтому все больше людей пытается освоить Hadoop и другие технологии Большых Данных.

Однако создание рабочей среды Hadoop — нетривиальная задача, требующая значительных усилий, особенно для новичков. Для решения этой проблемы Cloudera, являющаяся основным игроком в технологиях Большых Данных, создала на базе дистрибутива CentOS виртуальную машину (ВМ) QuickStart. Эта ВМ включает прекомпилированный стек Hadoop и отличный набор приложений Hadoop, и на нашем уроке мы поможем вам скачать эту ВМ, установить ее и начать использовать.

ВМ Cloudera QuickStart строится на 64-битном дистрибутиве Linux CentOS 6.4. Она бесплатна, и ее можно скачать в виде сжатого ZIP-файла, содержащего файлы OVF (Open Virtualization Format) и VMDK. Вы можете использовать VMware, KVM или VirtualBox,



› Импорт ВМ Cloudera QuickStart в VirtualBox очень прост. При импорте файла OVF нажмите на иконку файла, чтобы выбрать расположение файла, который вы извлекли из ZIP.

чтобы импортировать скачанные файлы и запустить ВМ. Мы используем VirtualBox, но вы можете выбрать любую другую программу виртуализации, на свой вкус. Учтите, что для запуска ВМ Cloudera потребуется VirtualBox версии 4 или новее. Скриншоты в данном руководстве сделаны в VirtualBox 5.0.8. А если вы хотите запустить Cloudera Manager, готовьте минимум 8 ГБ ОЗУ.

Прежде всего, скачайте и установите VirtualBox ([www.virtualbox.org](http://www.virtualbox.org)) для Linux с помощью команд YUM, APT или RPM, в зависимости от вашего дистрибутива. Установка на ваш компьютер VirtualBox не отличается от установки любого другого ПО. Также убедитесь, что вы используете 64-битную версию ОС хоста, и что под ВМ Cloudera QuickStart можно отвести как минимум 4 ГБ ОЗУ. Затем загрузите ВМ (<http://bit.ly/ClouderaDownloads>), выбрав ссылку QuickStart VM. По нажатию на ссылку скачивания у вас появится возможность выбрать виртуальную среду (VMware, KVM или VirtualBox). Выберите файл ZIP для VirtualBox; это может занять некоторое время, в зависимости от пропускной способности вашего интернет-соединения, поскольку он весит более 4 ГБ. Когда скачивание завершится, распакуйте файл, чтобы извлечь файлы VMDK и OVF. Теперь запустите VirtualBox и используйте опцию Import Appliance [Импорт устройства] в меню File, чтобы импортировать ВМ в VirtualBox (как на рис. вверху). В процессе импортирования вам придется указать путь к файлу OVF.

Указав его, нажмите кнопку Continue [Продолжить]. Оставшаяся часть установки представляет собой сплошные нажатия на кнопки Next [Далее] или Continue. Вам не надо менять никакие параметры по умолчанию или создавать/изменять какую-либо настройку. Все уже было преднастроено для вас. По умолчанию виртуальной машине будет выделено 4 МБ ОЗУ и один виртуальный CPU. Когда установка завершится, вы сможете запустить ВМ

## Скорая помощь

Вам понадобятся имя пользователя и пароль, чтобы войти в приложения, объединенные с ВМ Cloudera. При необходимости используйте cloudera как имя пользователя и пароль.

## Что такое Большие Данные?

Для многих компаний данные — это «валюта будущего» и существенный элемент, обеспечивающий конкурентоспособность, возможность расширить бизнес и стать лидером в своей отрасли.

Хранение, обработка и принятие решений, основанные на больших массивах данных, требуют сдвига парадигмы в области технологий. Мы больше не можем полагаться на централизованные реляционные базы данных, где данные хранятся и обрабатываются в одном месте. Данные стали слишком объемны для того, чтобы собирать и хранить их в одном месте. Так что вместо сбора данных в неком центре мы распределяем их и переносим

обработку туда, где данные хранятся. Технологии Больших Данных разработаны для управления и обработки больших по объему и распределенных данных с помощью стандартного оборудования и сетей.

Технологии Больших Данных — сердцевина всех крупных интернет-компаний, включая Google, Facebook, Twitter, Yahoo, LinkedIn и другие. Новые технологии, например, Интернет Вещей (IoT), потребуют от приложений Больших Данных анализа больших объемов информации, собранной с разных типов устройств.

Hadoop — это ведущая и очень популярная технология Больших Данных. Ее популярность привела

к взаимозаменяемости терминов Большие Данные и Hadoop в контексте распределенного хранения и обработки больших объемов данных.

У Hadoop имеются два основных компонента: MapReduce Framework и Hadoop File System (HDFS). На этих основных компонентах создано множество приложений и распределенных баз данных. Главное — помнить, что Большие Данные являются не единой технологией, а скорее экосистемой, состоящей из нескольких технологий, объединенных с целью хранения, анализа и оповещения об исключительно больших объемах распределенных данных.

из *VirtualBox*. Когда машина запустится, вы перейдете к Cloudera Desktop (как изображено на рис. внизу слева). Если web-браузер *Firefox* установлен в качестве браузера по умолчанию, вы, возможно, увидите окно *Firefox*.

## Файлы настройки Hadoop

При освоении новой системы мы всегда первым делом интересуемся, где находятся файлы настройки. Это помогает узнать, куда следует направляться для решения проблем или при желании пойти дальше и узнать что-то новое. В файлах настройки также содержатся важные подсказки по параметрам оптимизации производительности.

Прежде всего рассмотрим, как виртуальная машина инициирует сервисы Hadoop при запуске. Запустив следующую команду, вы увидите ряд скриптов запуска в директории */etc/init.d*. Обсуждение каждого из этих скриптов запуска находится за рамками нашего урока, но оно может дать вам идею, где можно найти больше информации, если у вас проблемы с запуском сервисов Hadoop или вы хотите узнать больше о запуске Hadoop:

```
$ ls -l /etc/init.d/hadoop*
-rwxr-xr-x 1 root root 4551 May 19 17:24 /etc/init.d/
hadoophdfs-datanode
-rwxr-xr-x 1 root root 4336 May 19 17:24 /etc/init.d/
hadoophdfs-journalnode
-rwxr-xr-x 1 root root 5065 May 19 17:24 /etc/init.d/
hadoophdfs-namenode
```

```
-rwxr-xr-x 1 root root 4402 May 19 17:24 /etc/init.d/
hadoophdfs-secondarynamenode
-rwxr-xr-x 1 root root 4886 May 19 17:24 /etc/init.d/hadoophttpfs
-rwxr-xr-x 1 root root 4423 May 19 17:24 /etc/init.d/
hadoopmapreduce-historyserver
-rwxr-xr-x 1 root root 4421 May 19 17:24 /etc/init.d/
hadoopyarn-nodemanager
-rwxr-xr-x 1 root root 4337 May 19 17:24 /etc/init.d/
hadoopyarn-proxyserver
-rwxr-xr-x 1 root root 4381 May 19 17:24 /etc/init.d/
hadoopyarn-Resourcemanager
```

Вам незачем вносить какие-либо изменения в эти файлы или включать/выключать какие-либо из этих сервисов: все они преднастроены за вас. Есть ряд других скриптов запуска в */etc/init.d* для других приложений Hadoop. В добавок к скриптам запуска можно найти файлы настройки для компонентов Hadoop в директории */etc*.

Cloudera прилагает к BM QuickStart ряд руководств, с пошаговыми примерами для освоения Hadoop. Домашняя страница web-браузера *Firefox* содержит ссылки на эти руководства (см. рис. внизу справа). Руководства Cloudera начинаются с бизнес-сценариев и используют утилиты Hadoop, доступные на виртуальной машине, чтобы воплотить их.

Первое руководство с примерами предполагает, что мы возьмем данные из реляционной базы данных и закинем их в распределенную файловую систему Hadoop (Hadoop Distributed File

## Скорая помощь

Если вы фанат командной строки, список команд Hadoop доступен на <http://bit.ly/HadoopCmdsGuide>.



Запустив BM Cloudera QuickStart, вы увидите рабочий стол по умолчанию. Новейшая версия BM заодно автоматически запустит браузер *Firefox*.

**cloudera LIVE**

Welcome to Your Cloudera QuickStart VM!

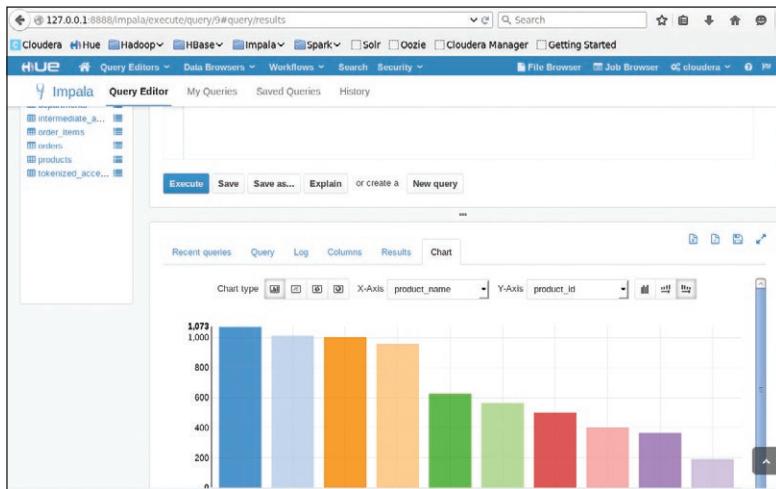
Node	Address
Manager Node	127.0.0.1
Worker Node 1	127.0.0.1

**Get Started**  
The tutorial below guides you through some analytic use cases, using the most popular open source tools included with CDH (including Cloudera Impala, Cloudera Search, and Hue).

**Analyze Your Data**

Домашняя страница web-браузера *Firefox* содержит ссылки на руководства Cloudera, которые дают отличный способ освоить комплект приложений.

**Подпишитесь на печатную или электронную версию** на [www.linuxformat.ru/subscribe](http://www.linuxformat.ru/subscribe)!



» Для запроса и отображения данных в различных форматах, включая легкие в построении диаграммы, можно использовать web-интерфейс *Hue* и запросы *Impala*.



**Функции Mapper и Reducer.** Типичное приложение Hadoop будет включать и функцию mapper, и функцию reducer. Вы можете написать обе функции на множестве языков, включая Java и Python.

System, HDFS). У типичного кластера Hadoop будет множество узлов для хранения проглоченных данных. Однако в этом сценарии вы запускаете только одну виртуальную машину, так что ваш «кластер» содержит только один узел данных. Обратите внимание, что независимо от количества информационных узлов в кластере, процесс будет один и тот же. Hadoop позаботится о распределении данных по нескольким узлам, так что вам не надо беспокоиться об этом.

Цель нашего урока — не пересказ примеров, предлагаемых Cloudera, а создание представления о том, чего ожидать от виртуальной машины. Не вдаваясь в подробности каждого примера, кратко перечислим имеющиеся примеры и их цели. Вы найдете подробное описание каждого из них по следующим ссылкам:

- Использование имеющихся данных, чтобы выяснить, какие продукты нравятся покупателям.
- Как использовать *Impala* для запроса данных с помощью web-интерфейса и как быстро создавать графики для визуализации результатов (см. на рис. вверху полосковую диаграмму, которую легко создать с помощью *Impala*). Можно выбрать и другие виды диаграмм, под свои потребности.
- Объединение логов web-доступа для оценки того, ведет ли самое большое число просмотров продукта к самому большому числу

продаж, например, оценка соотношения между количеством просмотров и данных по продаже продукта.

## ➤ Как анализировать с помощью Spark.

Есть и другие примеры, доступные в виртуальной машине, и вы можете изучить их во всех подробностях по мере улучшения навыков работы с технологиями Hadoop, и мы настоятельно рекомендуем вам изучить их все.

## Создание программы-счетчика слов

Теперь давайте напишем простую функцию Hadoop — mapper и reducer, которая будет считать слова в большом текстовом файле. В типичном сценарии мы бы рассчитывали на то, что файл будет распределен по многим информационным узлам. И Hadoop HDFS, и MapReduce framework будут управлять прозрачным запуском функций mapper и reducer во всех узлах. В нашем случае есть только один узел, так что вся обработка будет происходить в нем.

Вот список высокоуровневых шагов, предпринятых для этого примера:

1 Мы скачали большой текстовый файл (*2city10.txt*), являющийся текстом «Повести о двух городах» Чарльза Диккенса (этот файл доступен на <http://bit.ly/2city10>).

2 Мы создали директорию HDFS (*LXF\_input*), командой

```
hdfs dfs -mkdir /user/cloudera/LXF_input
```

3 Мы скопировали файл *2city10.txt* в директорию *LXF\_input* с помощью

```
hdfs dfs -put 2city10.txt /user/cloudera/LXF_input
```

Учтите, что если у нас несколько информационных узлов в кластере Hadoop, а это обычное явление, HDFS автоматически разделит и скопирует разные части файла в несколько узлов. Все это будет показано пользователю. В том и прелесть Hadoop, поскольку он автоматически управляет хранением данных на нескольких информационных узлах, создает несколько копий данных для резервирования и отслеживает, где хранятся данные.

4 Мы запускаем программы mapper и reducer для обработки данных с помощью потокового API Hadoop. И снова, если у нас несколько информационных узлов, Hadoop позаботится о запуске программы mapper на всех информационных узлах, где данные размещаются, перемешает и рассортирует промежуточные результаты и затем запустит программу reducer для создания конечного вывода. Конечные результаты хранятся в директории вывода, указанной в командной строке. Для этой цели используется следующая команда:

## Что входит в VM QuickStart?

Виртуальная машина QuickStart создана на дистрибутиве Linux CentOS 6.4. VM включает основные компоненты Hadoop (Hadoop Distributed File System, HDFS) и MapReduce Framework) и ряд вспомогательных приложений от Apache Foundation, а также от Cloudera. На виртуальной машине уже предустановлены все необходимые компоненты, так что вы можете включить VM и сразу начать работать, не проходя через процесс настройки. Последняя версия VM QuickStart также автоматически запускает Firefox, это поможет вам использовать множество инструментов прямо из web-браузера. В браузере предварительно настроена панель Favorites со всеми ссылками, которые понадобятся вам для обучения.

Как упоминалось ранее, основу Hadoop составляют среда HDFS и MapReduce. Поверх них прекомпилированная экосистема Больших Данных содержит ряд компонентов, включая, но не ограничиваясь следующим:

- **Сбор данных Hive** — утилита склада данных, а *Impala* — высокопроизводительный движок аналитики данных с открытым кодом.
- **Распределенная база данных HBase** — распределенная база данных, созданная поверх HDFS.
- **Импорт данных** От реляционных баз данных до Hadoop — *Sqoop* является утилитой, используемой для импорта данных в HDFS.
- **Взаимодействие с пользователем** Hadoop использует язык скриптов, а *Pig* — утилита, которая

поставляется с QuickStart для прямого взаимодействия пользователя с Hadoop, идущая со своим собственным языком скриптов, известным как PigLatin.

➤ **Web-интерфейс и визуализация** Hue обеспечивает интуитивный web-интерфейс для многих утилит, включая *Hive*, *Impala*, *Pig* и др.

Виртуальная машина Cloudera QuickStart также включает базу данных MySQL. Эта база данных поставляется с существующим хранилищем данных с образцами данных, которые вы можете импортировать в HDFS с помощью утилиты *Sqoop*. К виртуальной машине прилагается ряд руководств — они помогут вам приступить к работе с этими утилитами.

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

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar
-input /user/cloudera/LXF_input -output /user/cloudera/LXF_output
-mapper /home/cloudera/wordcount_mapper.py -reducer /home/
cloudera/wordcount_reducer.py
```

В этой командной строке мы начинаем с указания потокового API Hadoop; указываем директорию ввода, откуда читаются данные; директорию вывода, где должны храниться результаты; и пути к функциям mapper и reducer. Функции mapper и reducer написаны на Python и перечислены в шаге 5. Выполнение команды займет около минуты, и вы увидите статистику о работе и директории вывода.

По завершении предыдущей команды можете использовать следующую, чтобы увидеть файлы вывода в директории вывода:

```
$ hdfs dfs -ls /user/cloudera/LXF_output
Found 2 items [Найдено 2 объекта]
-rw-r--r-- 1 cloudera cloudera 0 2015-11-22 20:37 /user/cloudera/
LXF_output/_SUCCESS
-rw-r--r-- 1 cloudera cloudera 203900 2015-11-22 20:37 /user/
cloudera/LXF_output/part-00000
```

Обратите внимание, что файл вывода (**part-00000**) содержит результаты в виде пар ключ–значение. Пара ключ–значение покажет каждое слово в исходном файле и число его вхождений в файл. Чтобы посмотреть содержимое файла вывода, мы используем

```
hdfs dfs -cat /user/cloudera/LXF_output/part-00000
```

Поскольку файл достаточно длинен, можно направить вывод на команду more, чтобы прокручивать по страницам.

## Mapper и reducer

Функции mapper и reducer используют среду MapReduce для обработки данных. (Подробное обсуждение MapReduce на Apache см. по ссылке <http://bit.ly/HadoopApache>). Говоря по-простому, функция mapper создает пары ключ–значение. В указанном примере функция mapper читает каждую строку, разделяет строки на слова и затем помещает значение подсчета по слову, создавая пары ключ–значение.

Hadoop MapReduce «за кадром» перемешивает и сортирует эти пары и отправляет их функции reducer. Функция reducer читает каждую пару ключ–значение и объединяет пары, чтобы получить конечный результат.

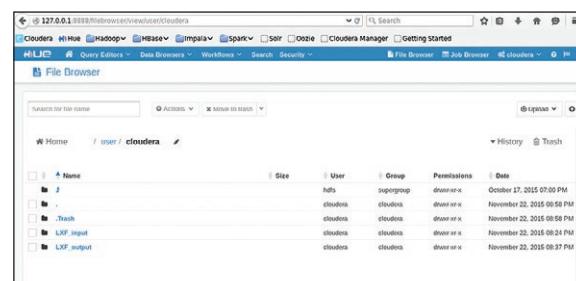
Давайте рассмотрим код этих функций. Тут для вас будет полезным некоторое знание языка Python, но если вам знаком хоть какой-нибудь другой язык программирования, вы должны вполне понять, как эти функции работают.

Далее идет код простой функции wordcount\_mapper.py. Функция mapper читает каждую строку текста, разделяет ее на слова (ключи) и присваивает значение каждому ключу.

```
#!/usr/bin/env python
import sys
for text_line in sys.stdin: # читаем все строки из стандартного
    #ввода
    text_line = text_line.strip() # удаляем переводы строки
    keys = text_line.split() #split line at blanks
    for key in keys: # присваиваем каждому слову значение
    #в цикле for
        value = 1
        print('{0}\t{1}'.format(key, value))
```

Reducer считает отсортированный вывод от MapReduce. Он читает пару ключ–значение в каждой строке и объединяет пары с совпадающими ключами, чтобы найти суммарное значение по каждому ключу. Потом он печатает итог в качестве вывода, который Hadoop сохраняет в HDFS-файле.

```
#!/usr/bin/env python
import sys
old_key = None
total_word_count = 0
```



Помимо множества опций, web-интерфейс Hue можно использовать для просмотра директорий и файлов HDFS.

```
for input_line in sys.stdin: # читаем все строки из стандартного
    #ввода
    input_line = input_line.strip() # удаляем переводы строки
    new_key, value = input_line.split("\t", 1)
    value = int(value) # преобразуем значение в целое число
    # если новый ключ равен старому, складываем значения
    # если новый ключ другой, печатаем пару ключ-значение
    if old_key == new_key:
        total_word_count += value
    else:
        if old_key:
            print(" {0}\t{1}".format(old_key, total_word_count))
        total_word_count = value
        old_key = new_key
    #print the last key-value pair
if old_key == new_key:
    print(" {0}\t{1}".format(old_key, total_word_count))
```

## Просмотр файлов HDFS

Создав образец файла mapper и reducer в HDFS, давайте разберемся, как воспользоваться web-интерфейсом Cloudera для просмотра файлов HDFS. Чтобы просматривать файлы и директории в web-браузере, авторизируйтесь в Hue (с помощью **username: cloudera** и **password: cloudera**) и затем нажмите на ссылку **File Browser** [Просмотр файлов] на голубой панели в верху web-страницы (рис. вверху). Вы увидите недавно созданную директорию **LXF\_input**, куда вы помещали текстовый файл для обработки, и директорию **LXF\_output**, созданную в результате выполнения функций `wordcount_mapper.py` и `wordcount_reducer.py`. Нажмая на названия этих директорий, вы можете более подробно изучить их изнутри, чтобы просмотреть отдельные файлы.

Есть две области, на которые вам стоит обратить внимание (см. скриншот вверху для справки): первая — панель Favorites, где вы увидите ссылки на Hue, Hadoop, HBase, Impala, Spark и другие утилиты. Они обеспечивает быстрый запуск этих приложений Hadoop, и Cloudera проделала отличную работу по включению всех важных ссылок в панель Favorites для легкого доступа.

Вторая область — это голубая панель Hue, появляющаяся сразу после входа в Hue. Ссылки на панели Hue позволяют проверить настройку, управлять пользователями Hue, создавать и редактировать потоки действий, запускать запросы и использовать другие инструменты, такие как Hive, Pig, Impala и пр. Вы также можете изобретать задания и просматривать статус отправленных заданий в web-интерфейсе.

Как вы, несомненно, сообразили, создание «рабочей» среды Hadoop — это, на самом деле, очень простая задача, пока вы используете нечто вроде BM Cloudera QuickStart для помощи. Однако это лишь начало вашего путешествия по океану Больших Данных. Мы бы предложили использовать руководство, находящееся на сайте Apache Foundation (<http://bit.ly/HadoopApache>), а также информацию на сайте Cloudera ([www.cloudera.com](http://www.cloudera.com)), чтобы понять, как работает MapReduce. Кроме того, примеры, представленные в BM Cloudera, будут очень полезны для лучшего понимания некоторых технологий. Удачи! **LXF**

ЧАСТЬ 3

# MySQL Fabric: Шардинг в деле

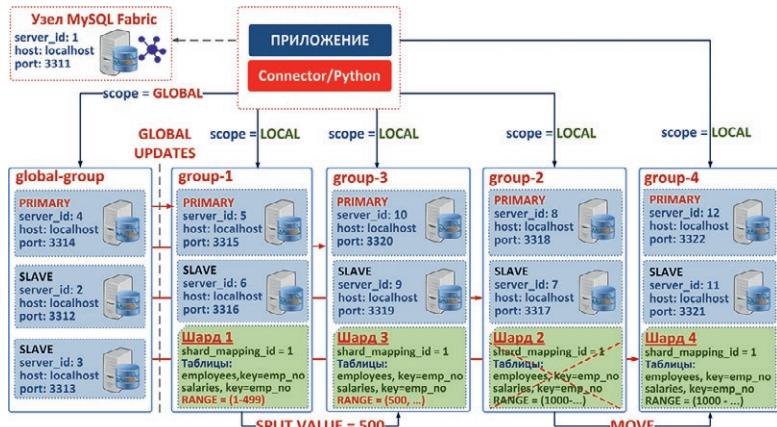
Под чутким руководством Лады Шерышовой займемся шардингом данных и масштабированием.



Наш  
эксперт

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

Рис. 1. Конфигурация тестовой фермы.



Шардинг в *MySQL Fabric* — это горизонтальное сегментирование, которое позволяет распределять данные, разбитые по некоторому принципу, между различными физическими серверами для обеспечения масштабируемости операций чтения/записи и облегчения обработки данных. Процесс шардинга — разнесение данных между отдельными шардами на основе заданного ключа, по выбранной стратегии. Причем каждый шард привязывается к отдельной высокодоступной (ВД) группе серверов. *MySQL Fabric* позволяет перемещать шарды на новые сервера и разделять их на множество других шардов. Для обеспечения высокой доступности шардинг в *MySQL Fabric* применяется совместно с репликацией данных.

В основе механизма шардинга данных, встроенного в *MySQL Fabric*, лежат следующие основные понятия:

» **Глобальная группа [global group]** ВД-группа, которая хранит все изменения схемы данных и распространяет их на все шарды. Сама по себе не является шардом.

» **Шард [shard]** Горизонтальный сегмент данных в таблице, определяемый значением ключа шардинга и принадлежащий определенной ВД-группе.

» **Ключ шардинга [sharding key]** Столбец таблицы, используемый для разбивки данных, размещенных в одной или более таблицах в заданной модели данных. Все множество ключей разбивается на диапазоны; для каждого диапазона назначается шард, где будут храниться данные из диапазона.

» **Стратегия шардинга** Определяет механизм формирования шардов и задается функцией HASH (хэш-секционирование) либо RANGE (секционирование по диапазонам). HASH предполагает равномерное распределение строк таблицы в зависимости от ключа шардинга, к которому в данном случае применяется HASH-функция. Эта стратегия удобна в случае строк с неравномерным или плохо группируемым распределением ключа шардинга.

» **Глобальная группа [global-group]** Узел *MySQL Fabric*, установленный и запущенный на локальной машине через порт 3311 (об этом — см. стр. 76 **LXF205-206**).

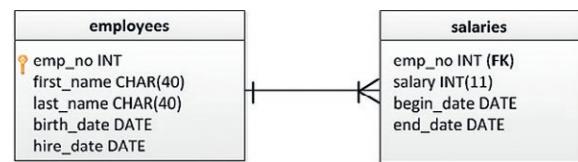


Рис. 2. Схема тестовой базы данных fabrictest.

RANGE распределяет данные по шардам на основе значений ключа шардинга, задаваемым для каждого шарда. В этом случае значение ключа описывается парой значений [min, max], где <min> — минимальное значение ключа в диапазоне, <max> — первое значение, превышающее диапазон. Каждому диапазону соответствует один шард.

» **Карта шардинга [shard mapping]** Описывает распределение данных между серверами в ВД-группах. Содержит набор таблиц — для каждой из них указывается столбец, который будет применяться как ключ шардинга, и стратегия шардинга.

Рассмотрим механизм шардинга данных в *MySQL Fabric* более подробно, на практике.

## Создаем и настраиваем ферму для тестирования

В конфигурацию тестовой фермы (рис. 1) входят:

- 1 Узел *MySQL Fabric*, установленный и запущенный на локальной машине через порт 3311 (об этом — см. стр. 76 **LXF205-206**).
- 2 Глобальная группа global-group, включающая сервера localhost:3312, localhost:3313 и localhost:3314 (PRIMARY).
- 3 ВД-группа group-1, состоящая из серверов localhost:3315 (PRIMARY) и localhost:3316.
- 4 ВД-группа group-2, включающая сервера localhost:3317 и localhost:3318 (PRIMARY).
- 5 ВД-группа group-3, включающая сервера localhost:3319 и localhost:3320 (PRIMARY); она будет использована для разделения шарда, принадлежащего группе group-1.
- 6 ВД-группа group-4, состоящая из серверов localhost:3321 и localhost:3322 (PRIMARY); в нее будет перемещен шард, принадлежащий группе group-2.

Все экземпляры *MySQL*-серверов устанавливаются в отдельные директории данных на одной и той же машине и запускаются на разных портах через свой сокет-файл под ОС Linux. Все сервера *MySQL* должны быть запущены с параметрами `--log_bin`, `--gtid-mode=ON`, `--enforce-gtid-consistency`, `--log_slave_updates`. Также для каждого сервера надо указать его уникальный идентификатор, в параметре `--server_id`.

Для создания экземпляра сервера *MySQL* выполняется команда  
`>> bin/mysqld --initialize --datadir=<MYSQL_DATA_DIR>`,  
 где `<MYSQL_DATA_DIR>` — путь к файлам данных.

```
linux-63ap:/usr/local/mysql # mysqlfabric group lookup_servers global-group
Fabric UUID: 5ca1abfe-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

server_uuid      address    status      mode weight
df5deao-08f1-11e5-abb3-0021634dd154 localhost:3312 SECONDARY READ_ONLY 1.0
f017416d-e8f1-11e5-acb3-0021634dd154 localhost:3313 SECONDARY READ_ONLY 1.0
fe32567b-e8f1-11e5-ae3c-0021634du154 localhost:3314 PRIMARY READ_WRITE 1.0

linux-63ap:/usr/local/mysql # mysqlfabric group lookup_servers group-1
Fabric UUID: 5ca1abfe-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

server_uuid      address    status      mode weight
0c49a3be-e8f2-11e5-ac41-0021634dd154 localhost:3315 PRIMARY READ_WRITE 1.0
d5b919d7-eb6a-11e5-9e07-0021634dd154 localhost:3316 SECONDARY READ_ONLY 1.0

linux-63ap:/usr/local/mysql # mysqlfabric group lookup_servers group-2
Fabric UUID: 5ca1abfe-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

server_uuid      address    status      mode weight
8ee80314-f74e-11e5-877c-0021634dd154 localhost:3317 SECONDARY READ_ONLY 1.0
9ecb8715-f74e-11e5-89ee-0021634dd154 localhost:3318 PRIMARY READ_WRITE 1.0
```

Рис. 3. Топология тестовой фермы после выполнения шагов 1 и 2.

Запуск MySQL-сервера осуществляется команда

```
>> bin/mysqlld_safe --server_id=<SERVER_ID> --log-bin --log-slave-updates --gtid-mode=ON --enforce-gtid-consistency
--datadir=<MYSQL_DATA_DIR> --port=<PORT_NUM>
--socket=<PATH>
```

где <SERVER\_ID> — уникальный идентификатор сервера, используемый для репликации данных; <PORT\_NUM> — номер порта; <PATH> — путь к сокет-файлу (например, /tmp/mysql1.sock).

Процесс установки и запуска MySQL-серверов подробно описан в LXF207 (стр. 72).

На каждом сервере, входящем в ВД-группу, необходимо создать 3-х пользователей с именами group (для доступа к серверу), fabric\_backup (для выполнения операций резервирования) и fabric\_restore (для выполнения операций восстановления) с указанием паролей, назначить им привилегии суперпользователей (GRANT ALL) и прописать их в конфигурационном файле **fabric.cfg**. Подробно процесс создания пользователей и настройки конфигурационного файла см. стр. 76 LXF205-206.

В качестве схемы данных будет создана база данных **fabrictest** и в ней — 2 таблицы: employees, содержащая информацию о сотрудниках, и salaries, содержащая данные об их зарплатах. Таблицы будут связаны по внешнему ключу — столбцу **emp\_no** (рис. 2).

## Создаем и настраиваем схему шардирования

### Создаем глобальную группу

Глобальная группа будет хранить все изменения структуры базы данных и распространять их на все шарды, входящие в схему шардирования. Согласно тестовой конфигурации, создадим глобальную группу с именем **global-group** и добавим туда 3 сервера:

```
>> mysqlfabric group create global-group --description='MySQL Fabric Global Group'
>> mysqlfabric group add global-group localhost:3312
>> mysqlfabric group add global-group localhost:3313
>> mysqlfabric group add global-group localhost:3314
```

Назначим группе главный сервер и подключим регистратор отказов для включения автоматического режима обнаружения сбоев:

```
>> mysqlfabric group promote global-group
>> mysqlfabric group activate global-group
```

### Создаем ВД-группы

Аналогичным образом создадим ВД-группы **group-1** и **group-2**, добавим туда по 2 сервера в соответствии с тестовой конфигурацией, назначим группам главные сервера и включим регистрацию отказов.

Результат выполнения шагов 1 и 2 представлен на рис. 3.

### Создаем карту шардирования

Определим стратегию, применяемую для шардинга, командой

```
linux-63ap:/usr/local/mysql # mysqlFabric sharding create_definition RANGE global-group
Fabric UUID: 5ca1abfe-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

uuid finished success result
50178c67-9a3b-49d3-9343-d555-5b6869f 1 1 2

state success when description
3 2 1459940886_43 Triggered by <mysql.Fabric.events.Event object at 0xa11242c>.
4 2 1459940886_56 Executing action (<_define_shard_mapping>).
5 2 1459940886_96 Executed action (<_define_shard_mapping>).
```

Рис. 4. Создание карты шардирования.

```
linux-63ap:/usr/local/mysql # mysqlFabric sharding list_definitions
Fabric UUID: 5ca1abfe-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

mapping_id type_name global_group_id
1 RANGE global-group

linux-63ap:/usr/local/mysql # bin/mysql --port=3311 --socket=/tmp/mysql1.sock -u fabric
-p -e "use fabric; select * from shard_maps;" Enter password:
+-----+
| shard_mapping_id | type_name | global_group |
+-----+
| 1 | RANGE | global-group |
+-----+
```

Рис. 5. Получение списка карт шардирования.

```
>> mysqlfabric sharding create_definition <TYPE_NAME> <GROUP_ID>,
где <TYPE_NAME> — стратегия шардирования (HASH или RANGE);
<GROUP_ID> — имя глобальной группы.
```

В нашем примере выберем функцию RANGE. Разделим данные в таблицах **employees** и **salaries** по диапазонам значений личных номеров сотрудников: (от 1 до 999) и (от 1000 и выше). Создадим карту шардирования (рис. 4), выполнив команду

```
>> mysqlfabric sharding create_definition RANGE global-group
Команда генерирует уникальный идентификатор карты шардирования (shard_mapping_id), и к нему в дальнейшем будут привязываться соответствующие таблицы базы данных. Посмотреть список карт шардирования можно либо выполнив команду >> mysqlfabric sharding list_definitions, либо выполнив запрос к таблице shard_maps хранилища состояния fabric, подключившись к узлу MySQL Fabric:
```

```
>> bin/mysql --port=3311 --socket=/tmp/mysql1.sock -u fabric -p -e "use fabric; select * from shard_maps;"
```

Результаты выполнения этих команд см. на рис. 5.

### Регистрируем таблицы в карте

Для определения соответствия между картой шардирования и объектами шардинга регистрируем таблицы базы данных и столбцы, составляющие ключ шардирования. Каждой шардируемой таблице сопоставляется уникальный идентификатор карты шардирования (shard\_mapping\_id), созданной на шаге 3. В нашем примере он равен 1. Таблицу в карту шардирования добавляет команда

```
>> mysqlfabric sharding add_table <SHARD_MAPPING_ID> <TABLE_NAME> <COLUMN_NAME>
где <SHARD_MAPPING_ID> — идентификатор карты шардирования, в которой регистрируется таблица, <TABLE_NAME> — имя шардируемой таблицы, <COLUMN_NAME> — наименование столбца таблицы, значение которого будет использоваться в качестве ключа шардирования.
```

&gt;&gt;

```
linux-63ap:/usr/local/mysql # mysqlFabric sharding lookup_table fabrictest.employees
Fabric UUID: 5ca1abfe-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

mapping_id type_name          table_name global_group column_name
1          RANGE              fabrictest.employees global-group emp_no

linux-63ap:/usr/local/mysql # mysqlFabric sharding lookup_table fabrictest.salaries
Fabric UUID: 5ca1abfe-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

mapping_id type_name          table_name global_group column_name
1          RANGE              fabrictest.salaries global-group emp_no

linux-63ap:/usr/local/mysql # bin/mysql --port=3311 --socket=/tmp/mysql1.sock -u fabric
-p -e "use fabric; select * from shard_tables;" Enter password:
+-----+
| shard_mapping_id | table_name | column_name |
+-----+
| 1 | fabrictest.employees | emp_no |
| 1 | fabrictest.salaries | emp_no |
+-----+
```

Рис. 6. Получение информации о таблицах, зарегистрированных в карте шардирования.

# Учебник MySQL Fabric

```
linux-63hp:/usr/local/mysql * bin/mysql --port=3311 --socket=/tmp/mysql1.sock
-u fabric -p -e "use fabric; select * from shards;"
```

Enter password:

shard_id	group_id	state
1	group-1	ENABLED
2	group-2	ENABLED

```
linux-63hp:/usr/local/mysql *
```

Рис. 7. Получение списка шардов.

В нашем тестовом примере будем шардировать 2 таблицы — employees и salaries, по столбцу emp\_no. Для регистрации этих таблиц в карте шардирования выполним такие команды (не забывая указывать имя базы данных, которой таблицы принадлежат!):

```
>> mysqlfabric sharding add_table 1 fabrictest.employees emp_no
>> mysqlfabric sharding add_table 1 fabrictest.salaries emp_no
```

Проверим результат регистрации таблиц с помощью команды lookup\_table:

```
>> mysqlfabric sharding lookup_table fabrictest.employees
>> mysqlfabric sharding lookup_table fabrictest.salaries
```

Это также можно сделать, обратившись к таблице shard\_tables хранилища состояния fabric, выполнив следующий запрос:

```
>> bin/mysql --port=3311 --socket=/tmp/mysql1.sock -u fabric -p
-e "use fabric; select * from shard_tables;"
```

Результаты выполнения этих команд представлены на рис. 6.

## 5 Добавляем шарды в схему шардирования

Для этого служит команда

```
>> mysqlfabric sharding <SHARD_MAPPING_ID> <GROUPID_LB_LIST> [-state=DISABLED]
```

<SHARD\_MAPPING\_ID> — идентификатор карты шардирования. <GROUPID\_LB\_LIST> — в случае использования функции RANGE задается список наименований групп <GROUP\_ID> с указанием нижних границ значений ключа <LOWER\_BOUND> в формате: "<GROUP\_ID>/<LOWER\_BOUND>, <GROUP\_ID>/<LOWER\_BOUND>...". В случае HASH-функции указывается только наименование группы <GROUP\_ID>. [-state=STATE] — указывает на то, включен (ENABLED) или отключен (DISABLED) шард. По умолчанию — DISABLED.

Согласно тестовой конфигурации, добавим в нашу схему шардирования 2 шарда, выполнив команду

```
>> mysqlfabric sharding add_shard 1 "group-1/1, group-2/1000"
--state=ENABLED
```

Если бы мы использовали функцию HASH, команда выглядела бы следующим образом:

```
>> mysqlfabric sharding add_shard 1 group-1 --state=ENABLED
```

Результат (рис. 7) проверяется запросом к таблице shards хранилища состояния Fabric:

```
>> bin/mysql --port=3311 --socket=/tmp/mysql1.sock -u fabric -p
-e "use fabric; select * from shards;"
```

```
linux-63hp:/usr/local/mysql * python mysqlfab/setup_table_gl_ha.py
linux-63hp:/usr/local/mysql * bin/mysql --port=3312 --socket=/tmp/mysql2.sock
-u group -p -e "use fabrictest; show tables;"
```

Enter password:

Tables_in_fabrictest
employees
salaries

```
linux-63hp:/usr/local/mysql *
linux-63hp:/usr/local/mysql * bin/mysql --port=3316 --socket=/tmp/mysql6.sock
-u group -p -e "use fabrictest; show tables;"
```

Enter password:

Tables_in_fabrictest
employees
salaries

```
linux-63hp:/usr/local/mysql *
linux-63hp:/usr/local/mysql * bin/mysql --port=3318 --socket=/tmp/mysql8.sock
-u group -p -e "use fabrictest; show tables;"
```

Enter password:

Tables_in_fabrictest
employees
salaries

Рис. 8. Проверка выполнения глобальных изменений схемы данных.

Как видим, в нашей схеме шардирования создано 2 рабочих шарда, расположенных в ВД-группах group-1 и group-2 и имеющих идентификаторы (shard\_id) 1 и 2 соответственно.

## 6 Настраиваем файл конфигурации Fabric

Для выполнения операций разделения и перемещения шардов необходимо в файле **fabric.cfg** в разделе [sharding] указать корректные пути к утилитам *mysqldump* (параметр *mysqldump\_program*) и *mysqlclient* (параметр *mysqlclient\_program*).

## Создаем схему данных в глобальной группе

Все операции над объектами базы данных и с самими данными будем выполнять через приложение, используя коннектор Python. Для этого создадим файл **setup\_table\_gl\_ha.py** и добавим туда код, приведенный ниже.

## 7 Создаем подключение к MySQL Fabric

Импортируем пакет *fabric* из библиотеки *Connector/Python* и прописываем параметры подключения к узлу *Fabric* соответственно его конфигурационному файлу:

```
import mysql.connector from mysql.connector
import fabric from mysql.connector
conn = mysql.connector.connect(
    fabric={"host": "localhost", "port": 32274, "username": "admin", "password": "admin", "report_errors": True},
    user="group", password="group", autocommit=True
)
```

Этот код необходимо будет вставлять в начало каждого последующего скрипта!

## 8 Задаем свойства подключения

Для выполнения глобальных изменений необходимо: указать список таблиц *tables*, над которыми будут выполняться изменения; область видимости *scope* установить в значение *fabric.SCOPe\_GLOBAL* (чтобы операции изменения объектов схемы данных выполнялись на всех шардах); режим доступа *mode* задать как *fabric.MODE\_READWRITE* (это говорит коннектору о том, что транзакция будет отправлена на главный сервер в группе).

```
conn.set_property(tables=["fabrictest.employees", "fabrictest.salaries"], scope=fabric.SCOPe_GLOBAL, mode=fabric.MODE_READWRITE)
```

## 9 Создаем объекты базы данных

Создадим базу данных *fabrictest* и в ней — две связанные внешним ключом таблицы *employees* и *salaries*:

```
cur = conn.cursor()
cur.execute("CREATE DATABASE IF NOT EXISTS fabrictest
CHARACTER SET utf8")
cur.execute("USE fabrictest")
cur.execute("DROP TABLE IF EXISTS salaries")
cur.execute("DROP TABLE IF EXISTS employees")
cur.execute(
    "CREATE TABLE IF NOT EXISTS employees (
        `emp_no` INT PRIMARY KEY AUTO_INCREMENT,
        `first_name` CHAR(40),
        `last_name` CHAR(40),
        `birth_date` DATE,
        `hire_date` DATE
    )")
cur.execute(
    "CREATE TABLE IF NOT EXISTS salaries (
        `emp_no` INT,
        `salary` INT(11),
        `begin_date` DATE,
        `end_date` DATE,
        FOREIGN KEY(`emp_no`) REFERENCES employees(`emp_no`)
    )")
```

)  
Запустим скрипт (>> python setup\_table\_gl\_ha.py) и проверим, что таблицы действительно создались на всех шардах. Выберем для проверки по серверу из каждой группы: localhost:3313 (global-group), localhost:3316 (group-1) и localhost:3317 (group-2). Подключимся к каждому из этих серверов и проверим список таблиц в базе данных:

```
>> bin/mysql --port=<PORT_NUM> --socket=<PATH> -u group -p -e "use fabrictest; show tables;"
```

Итак (рис. 8), база данных и таблицы создались на всех серверах.

## Операции обновления данных

Добавим записи в таблицы employees и salaries. Для этого создадим файл **add\_data\_gl\_ha.py**, с кодом для создания подключения к узлу MySQL Fabric. Определим процедуру вставки данных в таблицу employees:

```
def add_employee(conn, emp_no, first_name, last_name, birth_date, hire_date):
```

Зададим свойства подключения. Нобходимо указать: tables — шардированную таблицу или их список (через запятую), участвующих в операциях обновления данных; key — ключ шардирования; mode — режим обращения к серверу. Т.к. мы выполняем операции обновления данных, параметр mode д.б. fabric.MODE\_READWRITE.  
`conn.set_property(tables=["fabrictest.employees"], key=emp_no, mode=fabric.MODE_READWRITE)`  
`cur = conn.cursor()`  
`cur.execute(`  
`"INSERT INTO employees VALUES (%s, %s, %s, %s, %s)",`  
`(emp_no, first_name, last_name, birth_date, hire_date)`  
`)`

Определяем процедуру вставки данных в таблицу salaries:

```
def add_salary(conn, emp_no, salary, begin_date, end_date):
```

Аналогичным образом устанавливаем свойства подключения для таблицы salaries:

```
conn.set_property(tables=["fabrictest.salaries"], key=emp_no, mode=fabric.MODE_READWRITE)  

cur = conn.cursor()  

cur.execute(  

    "INSERT INTO salaries VALUES (%s, %s, %s, %s)", (emp_no,  

    salary, begin_date, end_date))
```

Вставим данные в таблицу employees, указав в свойствах подключения ее имя и область видимости fabric.SCOPE\_LOCAL:

```
conn.set_property(tables=["fabrictest.employees"], scope=fabric.SCOPE_LOCAL)  

add_employee(conn, 10, "Pavel", "Ivanov", "1972-10-01",  

    "2005-02-01")  

add_employee(conn, 30, "Elena", "Orlova", "1978-03-22",  

    "2008-04-15")  

add_employee(conn, 120, "Sergey", "Stepanov", "1982-08-09",  

    "2010-10-16")  

add_employee(conn, 550, "Olga", "Petrova", "1980-01-11",  

    "2011-11-14")  

add_employee(conn, 780, "Alex", "Plotnikov", "1979-05-18",  

    "2011-11-08")  

add_employee(conn, 1100, "Nataly", "Medvedeva", "1983-02-20",  

    "2012-06-10")  

add_employee(conn, 1330, "Egor", "Rogov", "1973-07-24",  

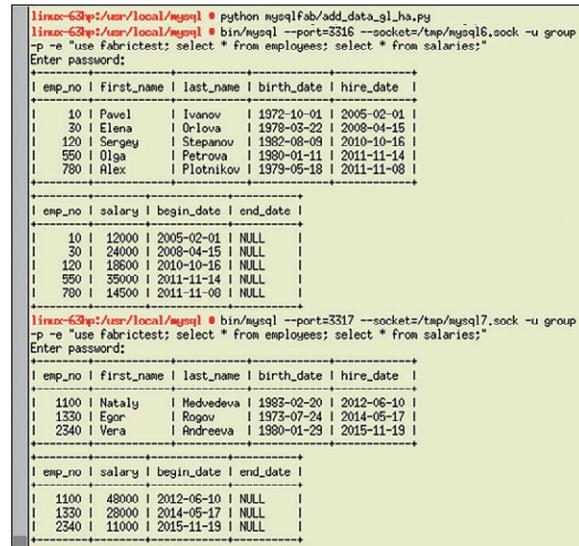
    "2014-05-17")  

add_employee(conn, 2340, "Vera", "Andreeva", "1980-01-29",  

    "2015-11-19")
```

Аналогичным образом вставляем данные в таблицу salaries, указав соответствующие свойства подключения:

```
conn.set_property(tables=["fabrictest.salaries"], scope=fabric.SCOPE_LOCAL)
```



```
linux-63ip:/usr/local/mysql * python mysqlfab/add_data_gl_ha.py  

linux-63ip:/usr/local/mysql * bin/mysql --port=3316 --socket=/tmp/mysql16.sock -u group -p -e "use fabrictest; select * from employees; select * from salaries;"  

Enter password:  

+-----+-----+-----+-----+-----+  
| emp_no | first_name | last_name | birth_date | hire_date |  
+-----+-----+-----+-----+-----+  
| 10 | Pavel | Ivanov | 1972-10-01 | 2005-02-01 |  
| 30 | Elena | Orlova | 1978-03-22 | 2008-04-15 |  
| 120 | Sergey | Stepanov | 1982-08-09 | 2010-10-16 |  
| 550 | Olga | Petrova | 1980-01-11 | 2011-11-14 |  
| 780 | Alex | Plotnikov | 1979-05-18 | 2011-11-08 |  
+-----+-----+-----+-----+-----+  
  
+-----+-----+-----+-----+  
| emp_no | salary | begin_date | end_date |  
+-----+-----+-----+-----+  
| 10 | 12000 | 2005-02-01 | NULL |  
| 30 | 24000 | 2008-04-15 | NULL |  
| 120 | 18600 | 2010-10-16 | NULL |  
| 550 | 35000 | 2011-11-14 | NULL |  
| 780 | 14500 | 2011-11-08 | NULL |  
+-----+-----+-----+-----+  
  
linux-63ip:/usr/local/mysql * bin/mysql --port=3317 --socket=/tmp/mysql17.sock -u group -p -e "use fabrictest; select * from employees; select * from salaries;"  

Enter password:  

+-----+-----+-----+-----+-----+  
| emp_no | first_name | last_name | birth_date | hire_date |  
+-----+-----+-----+-----+-----+  
| 1100 | Nataly | Medvedeva | 1983-02-20 | 2012-06-10 |  
| 1330 | Egor | Rogov | 1973-07-24 | 2014-05-17 |  
| 2340 | Vera | Andreeva | 1980-01-29 | 2015-11-19 |  
+-----+-----+-----+-----+-----+
```

```
add_salary(conn, 10, 12000, "2005-02-01", None)  

add_salary(conn, 30, 24000, "2008-04-15", None)  

add_salary(conn, 10, 12000, "2005-02-01", None)  

add_salary(conn, 30, 24000, "2008-04-15", None)  

add_salary(conn, 120, 18600, "2010-10-16", None)  

add_salary(conn, 550, 35000, "2011-11-14", None)  

add_salary(conn, 780, 14500, "2011-11-08", None)  

add_salary(conn, 1100, 48000, "2012-06-10", None)  

add_salary(conn, 1330, 28000, "2014-05-17", None)  

add_salary(conn, 2340, 11000, "2015-11-19", None)
```

Запустим скрипт — >> python add\_data\_gl\_ha.py — и проверим результат, выполнив запросы к таблицам employees и salaries на серверах localhost:3316 группы group-1 и localhost:3317 группы group-2 (рис. 9). Как видим, шард 1 (расположенный в группе group-1) содержит данные согласно указанному в карте шардирования диапазону личных номеров сотрудников от 1 до 999, а шард 2 (в группе group-2) — данные в диапазоне от 1000 и выше.

А как шардированные данные распределяются по серверам в ВД-группах, можно посмотреть, выполнив команду

```
>> mysqlfabric sharding lookup_servers <TABLE_NAME>  
<KEY_VALUE>
```

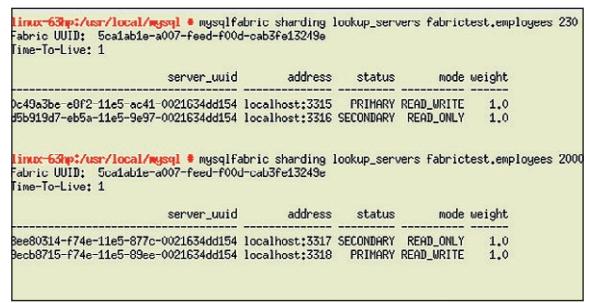
где <TABLE\_NAME> — наименование шардированной таблицы, <KEY\_VALUE> — значение ключа шардирования.

Выполним команды, указав таблицу employees и следующие значения ключа:

```
>> mysqlfabric sharding lookup_servers fabrictest.employees 230  

>> mysqlfabric sharding lookup_servers fabrictest.employees 2000
```

Как видим (рис. 10), данные со значением ключа 230 будут храниться на серверах, входящих в группу group-2 и принадлежащих шарду 1 (диапазон от 1 до 499), а данные со значением ключа 2000 попадут в группу group-2 шарда 2 (диапазон от 500 и выше) в соответствии с заданной картой шардирования. 



```
linux-63ip:/usr/local/mysql * mysqlfabric sharding lookup_servers fabrictest.employees 230  

Fabric UUID: 5ca1d1c-a007-feed-f00d-cab3f613219e  

Time-to-Live: 1  

server_uuid address status mode weight  

0c49a3be-e0f2-11e5-ac41-0021634dd154 localhost:3315 PRIMARY READ_WRITE 1.0  

55b919d7-eb5a-11e5-9e37-0021634dd154 localhost:3316 SECONDARY READ_ONLY 1.0  
  

linux-63ip:/usr/local/mysql * mysqlFabric sharding lookup_servers fabrictest.employees 2000  

Fabric UUID: 5ca1d1c-a007-feed-f00d-cab3f613249e  

Time-To-Live: 1  

server_uuid address status mode weight  

Bee80314-f74e-11e5-877c-0021634dd154 localhost:3317 SECONDARY READ_ONLY 1.0  

Becb8715-f74e-11e5-98ee-0021634dd154 localhost:3318 PRIMARY READ_WRITE 1.0
```

Рис. 10. Распределение шардированных данных по серверам.

Рис. 9. Разделение данных на шарды при выполнении операций обновления.

# IP-телефония: Завершаем наш диалплан

**Максим Черепанов** наносит последние штрихи в своей серии статей и отпускает вас в свободное плавание...



Наш  
эксперт

**Максим Черепанов** — заместитель директора в небольшой коммерческой фирме. По образованию железнодорожник-управленец. Linux занимается с 2008 г. Чем дальше это делает, тем дальше от Windows. За эти годы разучился чистить реестр и забыл слово «антивирус».

Приветствуя всех астерискостроителей на очередном уроке по VoIP-телефонии! В прошлый раз мы остановились на получении от сервера голосового сообщения по электронной почте. Продолжим готовить кубики для нашего первого диалплана.

## Голосовые файлы

Предположу, что вы выбрали способ записи звука с микрофона. Как вы помните, нам надо иметь несколько звуковых файлов:

- 1 «Привет! Вы позвонили в журнал Linus Format».
- 2 «Если вы хотите оставить голосовое сообщение, нажмите один. Если вы знаете номер абонента, наберите его в тоновом режиме или дождитесь ответа секретаря».
- 3 «Говорите в течение пятнадцати секунд после сигнала».
- 4 «Спасибо. Ваше обращение записано».
- 5 «До свидания».
- 6 «К сожалению, секретарь сейчас недоступен; пожалуйста, перезвоните позже. До свидания».

Почему именно этот набор, станет понятно чуть позже; просто надиктуйте эти фразы на микрофон при включенной записи в *Audacity*. Я записываю с частотой 22 кГц, затем после обработки снижаю до 8 кГц.

Итак, вы надиктовали фразу и остановили запись. Прежде всего обрежьте ненужные начало и хвост, перед первым звуком и после последнего. Затем избавьтесь от шума — для этого есть фильтр Эффекты > Удаление шума. Делается в 2 этапа: первым этапом выделяете паузу между словами и создаете модель шума. Затем применяете модель шума ко всему файлу. Попробуйте прослушать — скорее всего, скружутся все шипящие звуки. В этом нет ничего страшного: они всё равно уйдут за диапазон, который поддерживает *Asterisk*. Далее сделайте там же нормировку сигнала (выровняйте максимальный уровень до 0 дБ) и компрессию с установками по умолчанию. Таким образом вы приведете все

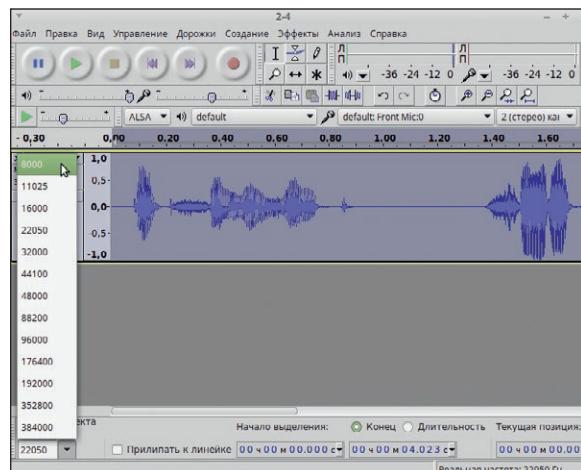


Рис. 2. Изменение частоты дискретизации.

звуки к уровню и динамическому диапазону, в которых наиболее чувствительно человеческое ухо. Если вы будете делать много таких файлов, то не лишнее будет создать цепочку обработки: Файл > Изменить цепочки, добавьте новую и внесите в неё нормировку и компрессию (рис. 1).

Обратите внимание: шумоподавление я в цепочку не включил, хотя оно напрашивалось само собой. Но у каждого файла своя модель шума, даже с учетом использования одного микрофона: вы можете говорить громче или тише, по улице может проехать автомобиль, кто-то за стеной хлопнет дверью. Поэтому шумы удалять надо из каждого файла отдельно.

Иное дело нормализация и компрессия — они одинаковы для всех файлов, их параметры вы можете видеть (и даже вручную изменить) в редактировании цепочки.

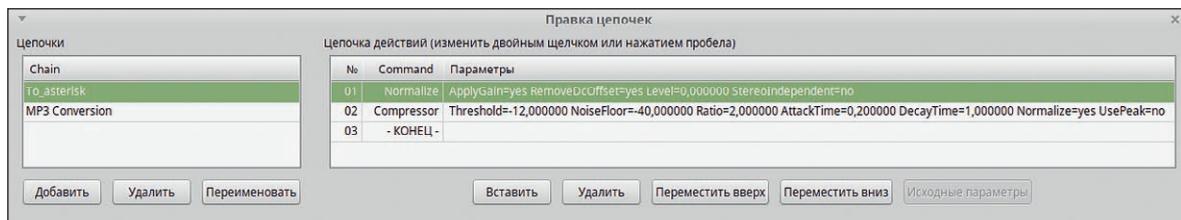


Рис. 1. Обработка звукового файла.

Потом вы сможете к каждому файлу просто применять этот шаблон. Поверьте, времени сэкономите уйму.

Конечный штрих: Создание > Тишина, добавьте в начало и конец файла по 1 секунде. Тогда ваш абонент нормально расслышит все слова, и ничего не смажется при переключении диалплана.

И самое главное — не забудьте поменять частоту! Посмотрите на рис. 2, это архиважно сделать.

Вы можете пропустить любой этап обработки, или их все — кроме приведения частоты к 8000 Гц. Иначе звука просто не будет!

После этого куда-нибудь экспортируйте файл, в формате 16-битного WAV.

Допустим, этот этап позади, и в какой-то папке вы имеете набор файлов от **1.wav** до **6.wav**. Загрузите №№ 1, 2 и 6 в *Asterisk*, как описано в предыдущей части (*FreePBX* в Admin > System Recordings, ищем надпись *Alternatively, upload a recording in any supported asterisk format...*). Имена файлам внутри *Asterisk* даем такие же, как и имена самих файлов (1, 2 и 6), дабы не возникло путаницы.

Теперь пора познакомиться с понятием *Announcement* — приветствие, объявление. Во время его действия *Asterisk* проигрывает звуковой файл, который либо можно, либо нельзя пропустить (как решит администратор), и после него осуществляется переход на другое действие диалплана.

Давайте сотворим из наших системных записей потребное количество приветствий. В дело пойдут не все сообщения: № 2 оставим для IVR-меню, №№ 3, 4 и 5 нам потребуются для прямой замены. Остались 1 и 6. Идем в Applications > Announcement и добавляем новое приветствие (рис. 3).

У меня отмечен пункт *Allow Skip*, это позволит при необходимости прокинуть приветствие. И по окончании приветствия направляем диалплан на отключение абонента (пока). Делаем так же для файла 6, и получаем две заготовки для диалплана.

Позволю вам напомнить цитату из предыдущей статьи, посмотрите на часть лог-файла при направлении абонента на голосовую почту. Там упоминаются файлы **vm-intro.gsm** и **auth-thankyou.gsm**. Это ссылки на стандартные файлы *Asterisk*, которые идут в комплекте с установкой.

Переименуйте свои заготовки: № 3 — в **vm-intro.wav**, № 4 — в **auth-thankyou.wav**, № 5 в **goodbye.wav**. И замените три системные записи *Asterisk*'а своими, они находятся в папках сервера **/var/lib/asterisk/sounds/en** и **/var/lib/asterisk/sounds/ru**.

С этого момента голосовая почта будет комментироваться по-русски (в части записи сообщения). Заодно посмотрите, насколько богат набор уже существующих слов и фраз в *Asterisk*'е. Если как следует там покопаться, то можно составить из стандартных средств вполне приличное IVR-меню вместе с приветствиями. Но если вам нужно нечто нестандартное, то все равно придется записывать самостоятельно (например, название вашей фирмы или должности работников для прямого набора). А изменение голоса при переходах выглядит непрофессионально и будет отвлекать от содержания фраз.

Ну вот, всё готово для главного действия.

## Построение IVR-меню

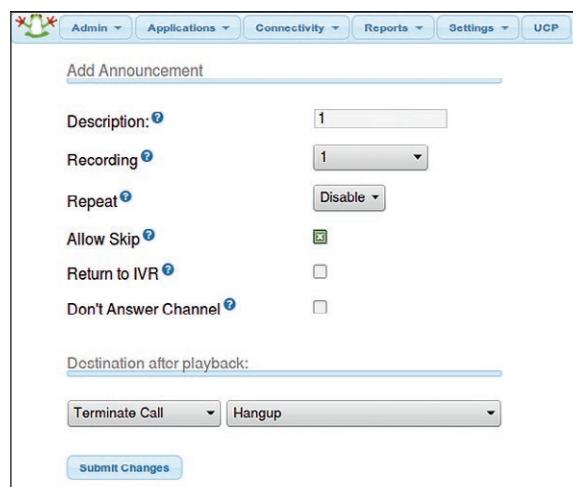
Идем в Applications > IVR; жмем кнопку Add a new IVR и начинаем заполнять окошки. Комментировать буду только то, что заполняю. Если не пишу ничего — оставляем пустое или то, что прописано по умолчанию.

» IVR Name — произвольно, я написал **main**.

» Announcement — выбираем свою запись 2.

» Direct Dial — поменяем на **Extensions**, это позволит, находясь в меню, набирать сразу те номера, которые есть на *Asterisk*'е.

» Timeout — ставим 4. Это время ожидания, когда *Asterisk* еще не переходит после окончания голосового файла на следующее действие, т. е. время на раздумье. Из нашего техзадания это 4 секунды, помните?



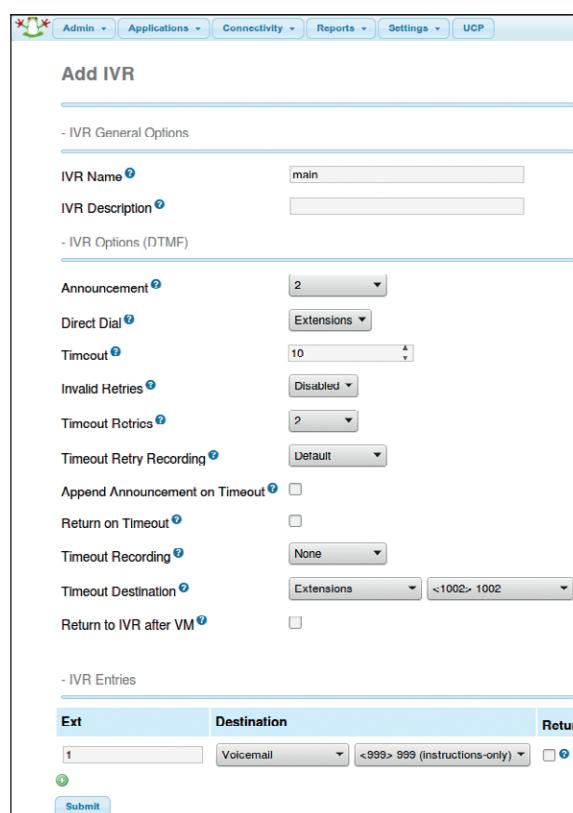
» Рис. 3. Добавление приветствия.

- » Invalid Retries — Disabled, отключим переход по неверному нажатию клавиш, хотя иногда это бывает очень полезно. После отключения у вас сразу уменьшится количество заполняемых полей, это нормально.
- » Timeout Retries — ставим 2, количество повторов голосовой записи.
- » Timeout Recording — None.
- » Timeout Destination — Extensions — <1002> 1002, если пользователь ничего не нажал, то направим его на номер 1002 (отрабатываем техзадание!).

Функционально у нас предусмотрена только одна клавиша — «1», и на нее надо повесить функцию записи голосового сообщения. Давайте вспоминать, что за это у нас отвечает номер 999.

С учетом того, что мы залили свои сообщения, вполне можно дать возможность *Asterisk*'у ими пользоваться. Поэтому там, где начинается поле IVR Entries, вводим в левое поле (Ext) 1, в следующем (Destination) сначала выбираем Voicemail — у вас сразу появляется еще одно окошко справа, и там уже стоит пользователь

»



» Рис. 4. Новое IVR-меню.

» Рис. 5. Входное приветствие.

999. Здорово, правда? Чуть подправим, выберем опцию <999> 999 (instructions-only).

Таким образом, ваше IVR-меню должно принять вид, показанный на рис. 4.

На мой взгляд, все очень просто и элегантно.

Как всегда, кнопочка Submit, и красная вверху Apply Config.

Но не расслабляемся: первая паутинка еще не готова, это внутреннее кружево, надо его на чем-то подвесить.

Фраза «Привет! Вы позвонили в журнал Linux Format» записана у нас в файле № 1, так же мы обозвали приветствие. После него нам надо перенаправить на только что сделанное меню.

Возвращаемся в Applications > Announcement, выбираем приветствие № 1, и внизу, в окошке Destination after playback: выбираем IVR main (рис. 5).

Сохраняем, применяем.

Далее нам надо направить на приветствие № 1 все наши входящие маршруты. Переходим в Connectivity > Inbound Routes, поочередно выбираем все наши маршруты, и внизу, в поле Set Destination выбираем наше приветствие (рис. 6).

И осталось совсем немного до реализации всех требований техзадания, всего один пункт: «При направлении на секретаря, если оператор занят или не отвечает, то перед абонентом надо извиниться: „К сожалению, секретарь сейчас недоступен; пожалуйста, перезвоните позже. До свидания!“ — и положить трубку». За этот фрагмент у нас отвечает звуковой файл № 6, который связан с одноименным приветствием (хотя в данном случае его более правильно назвать «прощание»).

Заходим в Applications > Extension, в наш номер секретаря (1002), и в последней части настроек направляем при всех условиях, когда он не отвечает, на «прощание с извинениями», это запись № 6 (рис. 7).

Submit, Apply Config — и вуаля, все готово! Можно звонить и проверять.

Мы с вами сейчас сделали самое простое, что можно придумать. Но уже какая автоматизация работы! Если возможности

» Рис. 7. Окончание работы.

ответить на звонок у вас нет — информация все равно вас догоит. И у вашего абонента не возникнет ощущение заброшенности и ненужности: ему оказали внимание и выслушали.

## Исходящие маршруты

Пора поговорить о том, как можно осуществлять звонки не только внутри Asterisk'a, но и звонить наружу — на городские или на сотовые телефоны.

К счастью, это проще всего того, что мы с вами уже проделали. И если бы у вас был всего один транк, то описание всего процесса уместилось бы на нескольких строчках. Но мы же хотим сделать умную маршрутизацию? Поэтому наберемся немногого терпения, и приступим.

За исходящие маршруты отвечает раздел Connectivity > Outbound Routes. У вас там уже есть два маршрута: один — это ТФОП, второй — сотовый модем. Добавим еще один:

» Connectivity > Outbound routes > ADD route

» Route Name > sip

» Dial Patterns that will use this Route > в 3-м слева поле ставим точку, это значит «любое количество символов»; для начала будем передавать в сеть так, как мы набираем на клавиатуре,

» Trunk Sequence for Matched Routes > выбираем свой SIP-транк.

При описании добавления исходящих маршрутов я не упоминал поле Route Position, здесь можно задавать приоритет используемых маршрутов. Если с Asterisk'a планируется звонить только в определенные направления (в город, на сотовые, по межгороду), то этого вам вполне хватит.

В реальности такого не бывает, звонить надо на разные номера. Соответственно, стоимость звонков по разным каналам будет разная. На сотовые телефоны выгоднее звонить с сотовых (желательно того же оператора, но это достаточно сложно реализовать), на городские — с того номера, где исходящие уже включены в абонентскую плату, по межгороду — там, где подключен наиболее правильный выбор межрегионального оператора.

Получается, что в данном случае нам надо также составить себе краткое техзадание на исходящую маршрутизацию. Для его составления нужно знать стоимость звонков. И на этой основе составить таблицу приоритетов — например, такую:

	SIP	ТФОП	Сотовый
Городской	2	1	3
Межгород	1	3	2
Сотовый	2	3	1

В первой колонке — куда звонить, в первой строке — транки, в самой таблице — стоимость звонков (чем меньше за минуту, тем приоритет выше, а порядковый номер меньше)

Т. е. Asterisk должен при наборе, например, сотового номера стараться соединиться через USB-модем. Если этот транк занят, то попытаться набрать через SIP-транк, и в последнюю очередь — через ТФОП.

Теперь надо дать понять Asterisk'у, какой из номеров является сотовым, какой городским и какой международным.

Сотовый номер состоит из 10 цифр, первая начинается на 9; т. е. любой сотовый номер можно записать так: 9XXXXXX XXXX, где

» Рис. 6. Направление маршрута.

X — любая цифра от 0 до 9. Теоретически первые 3 цифры, включая 9, определяют оператора мобильной связи. К сожалению, так бывает не всегда.

Городской (у меня) состоит из 6 цифр: XXXXXX.

Междугородный состоит из 10 цифр, и первая из них не 9: [1–8] XXXXXXXXX; возможно, к нему придется спереди добавлять цифру 8 (выход на межгород). Запись 1–8 в квадратных скобках означает любую цифру от 1 до 8.

Я намерено не упоминаю международный набор, короткие номера сотовых операторов, бесплатные телефоны горячих линий (8-800), телефоны экстренных служб (01, 02, 03, 112...). Для примера достаточно трех указанных выше вариантов, с остальными по аналогии разберетесь сами.

Теперь надо определить, как набирать эти номера: надо ли добавлять спереди к сотовым номерам цифру 8? Надо ли при звонке через сотового оператора в город добавлять внутренний код города?

Для этого просто поочередно отключайте все транки, кроме одного, и пытайтесь дозвониться разными вариантами — то в город, то на сотовые, то на межгород. Смотрите при этом в консоль Asterisk'a и делайте выводы. Например, я пытаюсь дозвониться на свой сотовый с внутреннего номера 1002 через SIP-транк:

```
-- Executing [s@macro-dialout-trunk:22]
Dial("SIP/1002-00000015", "SIP/78634431XX/90343544XX,300,
Tt") in new stack
```

Но не указал спереди 8, поэтому в итоге оператор сообщил мне, что номер не обслуживается, и разорвал канал:

```
[2016-03-17 13:39:06] WARNING[6810][C-0000000f]:
channel.c:4861 ast_prod: Prodding channel 'SIP/1002-00000015'
failed
== Spawn extension (macro-dialout-trunk, s-INVALIDNMBR, 4)
exited non-zero on 'SIP/1002-00000015' in macro 'dialout-trunk'
== Spawn extension (from-internal, 90343544XX, 5) exited non-
zero on 'SIP/1002-00000015'
-- Executing [h@from-internal:1] Hangup("SIP/1002-00000015",
") in new stack
```

Тогда я прибавил восьмёрку, и вот что вышло:

```
-- Executing [s@macro-dialout-trunk:22]
Dial("SIP/1002-00000017", "SIP/78634431XX/890343544XX,300,
Tt") in new stack
```

И затем:

```
-- Called SIP/78634431XX/890343544XX
```

То есть понятно, что для вызова сотового телефона с SIP-транка надо прибавить к сотовому номеру 8, а набирать его через Asterisk без этой цифры. Или набирать с цифрой, но по второй сортировать вызовы на мобильные телефоны и на межгород (9 — мобильный, 1–7 — межгород). Остановимся пока на первом варианте.

По итогам тестовых звонков составляем вторую табличку:

	SIP	ТФОП	Сотовый
Городской	XXXXXX	9XXXXXX	8XXXXXXXXX
Межгород	8[1-8]XXXXXXXXXX	98[1-8]XXXXXXXXXX	8[1-8]XXXXXXXXXX
Сотовый	89XXXXXXXXXX	89XXXXXXXXXX	89XXXXXXXXXX

В колонке «ТФОП» впереди всех номеров стоит 9 — у меня выход в город за пределы офисной железной АТС (не Asterisk!) проходит через девятку. В колонке «Сотовый» XXXX — код моего города, только так соединяется через мобильного оператора. Разумеется, звонок через сотовый modem в город напоминает нечто «почесать левое ухо правой рукой», да это еще и дорого. Но зато при занятости основных каналов мы все равно сможем позвонить куда угодно. Тем более, что приоритет сотового модема при звонке в город равен 3.

Вооружившись этими табличками, приступаем к ваянию маршрутизации.

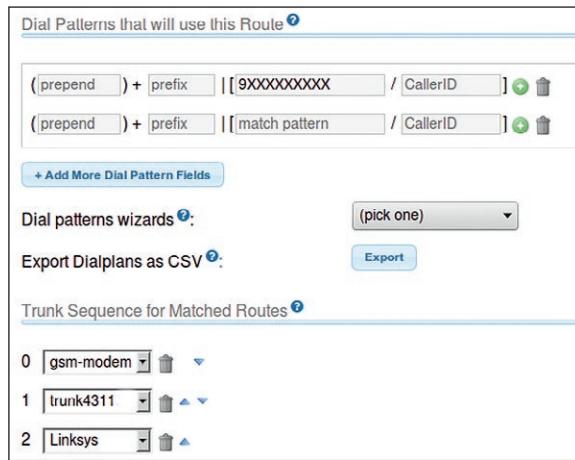


Рис. 8. Приоритеты в маршруте.

Перво-наперво стираем все старые исходящие маршруты, они нам больше не понадобятся.

Создаем три новых: city, intercity, mobile. В каждом из них в области Dial Patterns that will use this Route прописываем маску, по которой Asterisk будет определять, какой из номеров подходит под этот маршрут. Поля prepend и prefix оставляем пустыми, а в поле match pattern вносим маски:

- » city — XXXXXX
- » intercity — [1–8]XXXXXXXXXX
- » mobile — 9XXXXXXXXXX

Спускаемся ниже, и в поле Trunk Sequence for Matched Routes выбираем для каждого из маршрутов свою цепочку приоритетов, которую берем из таблички; например, маска и приоритеты для маршрута mobile показаны на рис. 8.

Переходим в редактирование транков: Connectivity > Trunks, и прописываем для каждого транка правила набора номеров, это делается в области Dialed Number Manipulation Rules. Допустим, это SIP-транк, как на рис. 9.

Первая строка — межгород, вторая сотовые, третья городские. Для канала ТФОП всюду в поле prepend в крайнюю левую позицию будут добавлены девятки (выход в город).

Пробуйте звонить с Asterisk'a на сотовые, городские и межгород — он безошибочно будет подставлять заданные вами приоритеты.

## Заключение

На этом я заканчиваю серию статей, которую справедливо было бы назвать «Asterisk+FreePBX для чайников: начало». Я уверен, что вы под глубоким впечатлением от возможностей IP-телефонии. Остается только направить вас на первоисточник: «Asterisk: The Future of Telephony [Астериск: Будущее телефонии]», авторы Джим Ван Меггелен, Лейф Мадсен, Джаред Смит [Jim Van Meggelen, Leif Madsen, Jared Smith]. В Интернете есть хороший перевод на русский язык — например, на [http://www.e-reading.club/bookreader.php/138011/Asterisk:\\_budushchee\\_telefonii.pdf](http://www.e-reading.club/bookreader.php/138011/Asterisk:_budushchee_telefonii.pdf). Там нет «родного GUI», но возможностей на порядок больше. Прочтите, не пожалеете. И устойчивой вам связи!

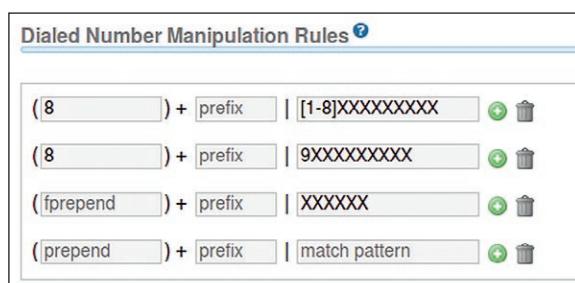


Рис. 9. Правила набора в транке.



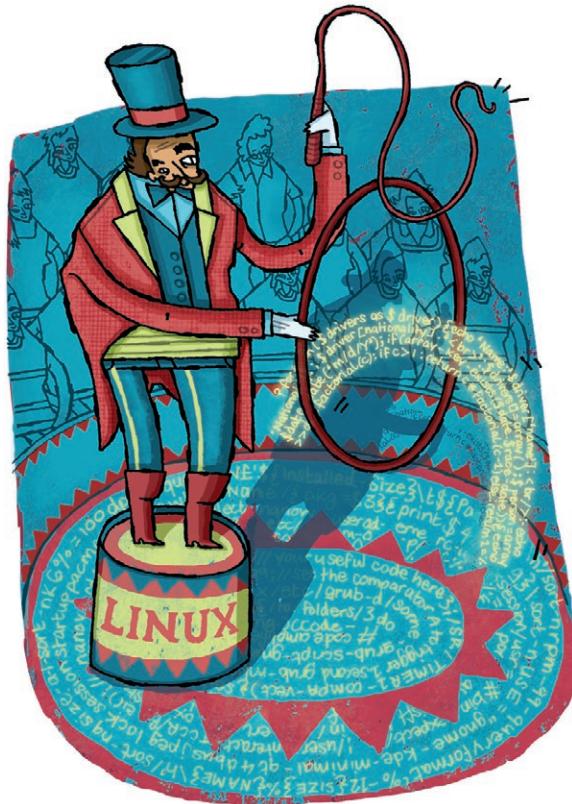
# Swift: Базовое руководство

Наберите обороты с языком программирования Swift — и научитесь программировать быстро. **Михалис Цукалос** разбирается в основах.



Наш  
эксперт

**Михалис Цукалос** — администратор баз данных, программист, математик и администратор Unix. В свободное время он любит писать технические статьи и сражаться с преступностью.



Главным сюрпризом Всемирной конференции разработчиков на платформе Apple (WWDC) 2014 г. было анонсирование Swift, нового языка программирования, шедшего на замену Objective-C от Apple. Главным сюрпризом WWDC 2015 было провозглашение Swift открытым проектом. Наше руководство представляет собой краткое введение в Swift и его возможности с помощью нескольких небольших, но полноценных примеров кода.

Процесс установки, скорее всего, покажется самой сложной частью этого урока. Apple не предоставляет пакетов для всех дистрибутивов Linux. Пока официально поддерживаются только Ubuntu 14.04 и 15.10. Если у вас другой дистрибутив, процесс установки скорее всего усложнится. Для Ubuntu 14.04 этот процесс таков:

```
$ sudo apt-get install clang libicu-dev
$ wget https://swift.org/builds/ubuntu1404/swift-2.2-SNAPSHOT-2015-12-18-a/swift-2.2-SNAPSHOT-2015-12-18-a-ubuntu14.04.tar.gz
$ wget -q -O - https://swift.org/keys/all-keys.asc | gpg --import -
$ gpg --keyserver hkp://pool.sks-keyservers.net --refresh-keys
Swift
```



Скобки в выражениях if не обязательны, так как, по мнению Swift, иногда они неоправданно загрязняют код. В Swift также нет файлов.h: они здесь не нужны.

```
motsouk@mtsouk-ubuntu:~ (esh)
mtsouk@mtsouk-ubuntu:~$ ll /home/mtsouk/swift-2.2-SNAPSHOT-2015-12-18-a-ubuntu14.04/usr/bin
total 145400
drwx----- 2 mtsouk mtsouk 4096 Dec 19 09:39 ./
drwx----- 7 mtsouk mtsouk 4096 Dec 19 09:39 ../
lrwxrwxrwx 1 mtsouk mtsouk 10 Dec 19 09:38 lldb -> lldb-3.8.0*
-rwx----- 1 mtsouk mtsouk 56959 Dec 19 09:36 lldb-3.8.0*
-rwx----- 1 mtsouk mtsouk 86318 Dec 19 09:36 lldb-objdumper*
lrwxrwxrwx 1 mtsouk mtsouk 13 Dec 19 09:38 lldb-mi -> lldb-mi-3.8.0*
-rwx----- 1 mtsouk mtsouk 927980 Dec 19 09:36 lldb-mi-3.8.0*
lrwxrwxrwx 1 mtsouk mtsouk 1024 Dec 19 09:38 lldb-server -> lldb-server-3.8.0*
-rwx----- 1 mtsouk mtsouk 63672187 Dec 19 09:36 lldb-server-3.8.0*
-rwx----- 1 mtsouk mtsouk 9177 Dec 19 09:35 repl_swift*
-rwx----- 1 mtsouk mtsouk 7308411 Dec 19 09:32 swift*
lrwxrwxrwx 1 mtsouk mtsouk 5 Dec 19 09:38 swift-autolink-extract -> swift*
-rwx----- 1 mtsouk mtsouk 1754089 Dec 19 09:39 swift-build*
-rwx----- 1 mtsouk mtsouk 7683691 Dec 19 09:36 swift-build-tool*
lrwxrwxrwx 1 mtsouk mtsouk 5 Dec 19 09:38 swiftc -> swift*
-rwx----- 1 mtsouk mtsouk 856388 Dec 19 09:31 swift-demangle*
mtsouk@mtsouk-ubuntu:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/usr/games:/usr/local/games
mtsouk@mtsouk-ubuntu:~$ export PATH="/home/mtsouk/swift-2.2-SNAPSHOT-2015-12-18-a-ubuntu14.04/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/usr/games:/usr/local/games"
mtsouk@mtsouk-ubuntu:~$ echo $PATH
/home/mtsouk/swift-2.2-SNAPSHOT-2015-12-18-a-ubuntu14.04/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/usr/games:/usr/local/games
Swift version 2.2 dev (LLVM 3ebdbb2c7e, Clang f66c5bb67b, Swift 0ddf238ad7)
Target: x86_64-unknown-linux-gnu
mtsouk@mtsouk-ubuntu:~$ swift
Welcome to Swift version 2.2.dev (LLVM 3ebdbb2c7e, Clang f66c5bb67b, Swift 0ddf238ad7)
. Type :help for assistance.
1>
```

Содержимое каталога bin в установке Swift.

```
$ wget https://swift.org/builds/ubuntu1404/swift-2.2-SNAPSHOT-2015-12-18-a/swift-2.2-SNAPSHOT-2015-12-18-a-ubuntu14.04.tar.gz.sig
$ gpg --verify swift-2.2-SNAPSHOT-2015-12-18-a-ubuntu14.04.tar.gz
```

Первая команда устанавливает ряд необходимых пакетов, если они еще не установлены. Вторая загружает последний снимок в версии для разработчиков. Третья импортирует ключи PGP и должна выполняться по разу в каждой системе Linux. Четвертая команда загружает последние ключи, а пятая — необходимые файлы подписи. Последняя команда проверяет целостность загруженного архива. Учтите, что последняя команда иногда выдает безобидное предупреждение, которое можно игнорировать, если вы загрузили все файлы из доверенного источника (например, <https://swift.org>).

Если все нормально, распакуйте архив командой

```
tar xzf swift-2.2-SNAPSHOT-2015-12-18-a-ubuntu14.04.tar.gz
```

Она создает новый каталог, который надо добавить к своей переменной PATH:

```
$ export PATH="/home/mtsouk/swift-2.2-SNAPSHOT-2015-12-18-a-ubuntu14.04/usr/bin":$PATH"
```

Примечание: приведите путь /home/mtsouk/swift-2.2-SNAPSHOT-2015-12-18-a-ubuntu14.04 в соответствие со своей системой, именем пользователя и каталогом Swift. Свою версию Swift можно определить, выполнив команду `swift --version`. (На экранном снимке вверху показано содержимое каталога bin установки Swift.)

## Другие свойства Swift

Значения в Swift никогда не преобразуются в другие типы неявно, что способствует уменьшению числа ошибок в вашем коде. Фигурные или круглые скобки ставятся всегда, даже для блоков кода, состоящих всего из одной строки. Как мы упоминали в основном тексте, наиболее мощная управляющая конструкция в Swift — это выражение `switch`. Поскольку Swift поддерживает функциональное

программирование, функция Swift может возвращать в качестве значения другую функцию! Кроме того, в именах переменных можно использовать любые символы Unicode. Например, можно объявить переменную так:

```
let π = 3.14159
```

Если переменная, объявленная с помощью ключевого слова `var`, ни разу не меняла своего

значения, Swift понимает это и выводит следующее предупреждение:

```
bubbleSort.swift:7:21: warning: variable 'temp' was never mutated; consider changing to 'let' constant [переменная 'temp' не менялась, ее можно объявить константой]
```

Swift также поддерживает выражения `while` и `repeatwhile`. Последнее похоже на `do-while` в C.

Запуск команды `swift` без аргументов откроет интерактивную оболочку Swift REPL. Это прекрасное место для изучения новых вещей, так как в ней вы сможете прочесть, вычислить и вывести результаты для любого введенного кода Swift. В REPL доступны все функции Swift. А если нажать клавишу Tab, REPL откроет список всех возможных дополнений для текста, который вы набрали. Начнем с представления маленькой программы "Hello World!" на Swift:

```
print("Hello, World!")
```

Если вы знакомы с C, C++ или Objective-C, то кое-что отмите. Во-первых, здесь нет точки с запятой, означающей окончания команд. В Swift точки с запятой используются только для того, чтобы поместить две команды в одну строку. Во-вторых, здесь нет функции `main()`, что должно быть для вас большим сюрпризом. Наконец, здесь нет выражений `#include` или `#import`; однако это вызвано лишь тем, что наша программа "Hello World" очень маленькая и она ей не нужны.

Запустим программу "Hello World!":

```
$ swift hw.swift
Hello, World!
$ swiftc hw.swift
$ ls -l hw
-rwxrwxr-x 1 mtsouk mtsouk 13676 Dec 24 18:19 hw*
$ ./hw
Hello, World!
$ cat hw.script
#!/home/mtsouk/swift-2.2-SNAPSHOT-2015-12-18-aubuntu14.04/
usr/bin/swift
print("Hello World!")
$ chmod 755 hw.script
$ ./hw.script
Hello World!
```

Как видите, запустить код Swift можно тремя различными способами. Первый — выполнить код Swift из командной строки без создания исполняемого файла. Второй способ использует `swiftc` для компиляции `hw.swift` и создания исполняемого двоичного файла, который автоматически получает имя `hw` — по имени файла с кодом Swift. Третий способ показывает, как создать и запустить скрипт Swift (`hw.script`) — по аналогии со скриптом оболочки Unix. Если вы успешно запустили программу "Hello World", можно двигаться дальше.

Apple включила в пакет Swift менеджер пакетов Swift и систему сборки (Build System), что позволяет удобно компилировать проекты Swift. Для примера воспользуемся файлом `hw.swift`. Сначала надо создать новый каталог, который будет включать все файлы проекта — для этого выполните команды `$ mkdir hw` и `$ cd hw`.

Затем надо создать файл с именем `Package.swift`, который будет манифестом проекта — это действие обязательное. Если файл манифеста пуст, менеджер пакетов создаст новый проект с параметрами по умолчанию. Так как вы используете параметры

по умолчанию, менеджер пакетов ожидает найти весь код Swift в каталоге `Sources`, который тоже надо создать:

```
$ touch Package.swift
$ mkdir Sources
```

По умолчанию менеджер пакетов будет искать файл `main.swift` и компилировать его, поэтому скопируйте файл `hw.swift` в каталог `Sources`, но при этом переименуйте этот файл в `main.swift` и затем соберите проект следующим образом:

```
$ cp ~/hw.swift Sources/main.swift
$ swift build
```

Результат сборки будет находиться в каталоге `.build/debug`. Это вызвано тем, что менеджер пакетов считает, что первый исполняемый файл будет использован для отладки. Имя исполняемого файла совпадает с именем проекта. (Полный список созданных для проекта файлов и каталогов см. на экранном снимке внизу.) В оставшейся части руководства мы продемонстрируем важные возможности языка, поэтому готовьтесь к написанию кода на Swift!

## Константы и переменные

В Swift надо объявлять и константы, и переменные. Вот как объявляются переменная и константа в REPL:

```
1> var youCanChangeMe = 12
youCanChangeMe: Int = 12
2> let youCannotChangeMe = 12
youCannotChangeMe: Int = 12
```

Для объявления переменной следует использовать ключевое слово `var`, для константы — ключевое слово `let`. Если вы попробуете изменить значение константы `youCannotChangeMe`, Swift разразится очень подробным сообщением об ошибке (как вы увидите



Две хорошие книги о Swift — «Программирование на Swift: Руководство для чайников» Мэттью Маттиаса [Matthew Mathias] и Джона Галлахера [John Gallagher] и «Новичок в Swift: Начинаем программировать на Swift 2» Джени Клейтон [Janie Clayton], Алексиса Галлахера [Alexis Gallagher], Мэтта Геллоуэя [Matt Galloway], Эли Ганема [Eli Ganem], Эрика Кербера [Erik Kerber] и Бена Морроу [Ben Morrow].

```
mtsouk@mtsouk-ubuntu:~/code/swift/hw$ ssh
:.
: ... .build Package.swift Sources
./.build:
: ...
./.build/debug:
: ...
./.build/debug/hw.o:
: ...
./.build/debug/hw.o/home:
: ...
./.build/debug/hw.o/home/mtsouk:
: ...
./.build/debug/hw.o/home/mtsouk/code:
: ... swift
```

Менеджер пакетов Swift, который входит в состав пакета Swift, создает при сборке проекта массу файлов и каталогов.

**» Подпишитесь на печатную или электронную версию** на [www.linuxformat.ru/subscribe](http://www.linuxformat.ru/subscribe)!



```
2. mtsouk@mtsouk-ubuntu: ~/code/swift (ssh)
mtsouk@mtsouk-ubuntu:~/code/swift$ swift
Welcome to Swift version 2.2-dev (LLVM 3ebdbb2c7e, Clang f66c5bb67b, Swift 0ddf238ad7)
· Type :help for assistance.
1> var youCanChangeMe = 12
youCanChangeMe: Int = 12
2> let youCannotChangeMe = 12
youCannotChangeMe: Int = 12
3> youCannotChangeMe = 23
repl.swift:3:19: error: cannot assign to value: 'youCannotChangeMe' is a 'let' constant
youCannotChangeMe = 23
~~~~~ ^
repl.swift:2:1: note: change 'let' to 'var' to make it mutable
let youCannotChangeMe = 12
~~~
var

3> youCan
Available completions:
    youCanChangeMe: Int
    youCannotChangeMe: Int
3> youCanChangeMe = 23
4> print(youCanChangeMe)
23
5> print(youCannotChangeMe)
12
6> youCanChangeMe = "But you cannot change its type!"
repl.swift:6:18: error: cannot assign value of type 'String' to type 'Int'
youCanChangeMe = "But you cannot change its type!"
~~~~~ ^
```

➤ Изменить значение константы, объявленной с помощью `let`, нельзя. Сделать это можно только с переменными, объявленными с помощью `var`. Также нельзя изменить тип переменной!

далее). Также нельзя изменять тип переменной, даже если она объявлена с ключевым словом `var`. Язык Swift достаточно умен, чтобы определить тип переменной автоматически. Но его можно объявить и самим, следующим образом:

```
var youCanChangeMe: Int = 123
```

В следующем коде Swift определяется рекурсивная функция, которая позволит вычислить целые числа, принадлежащие к последовательности Фибоначчи:

```
func fibo(number: Int) -> Int {
    if number == 0 {
        return number
    }
    if number == 1 {
        return number
    }
    return (fibo(number - 1) + fibo(number - 2))
}
```

Объявление функции `fibo()` показывает, что она принимает в качестве входного параметра целое число, присваиваемое переменной `number`, и возвращает тоже целое число. Если вы хотите вычислить первые 15 членов последовательности Фибоначчи, вам необходимо использовать следующий цикл `for`:

```
for i in 0...15 {
    print("Fibonacci number \((i) is: \(fibo(i))")
}
```

Использование `\()` в функции `print()` позволяет программисту включить переменную или возвращаемое значение функции в строку. Если вы знакомы с C, то видите, что указывать тип выводимой переменной не надо.

Тот же самый цикл `for` можно записать и по-другому:

```
let loop = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
for i in loop {
    print("Fibonacci number \((i) is: \(fibo(i))")
}
```

Последний простой пример показывает, что можно перебирать элементы массива с помощью цикла `for`. Наряду с новым стилем

цикла `for` в Swift можно использовать и старый, например: `for i = 0; i<=15; i++ {}`.

Однако в будущих версиях Swift C-версия цикла `for` может быть удалена, поэтому советуем вам использовать новую версию. Если вы не хотите включать в цикл правую границу, его нужно записать как `for i in 0..5 { }; а если хотите указать особый шаг, то как for i in 2..stride(to: 10, by: 2) {}.`

Предыдущий цикл `for` с помощью функции `stride()` возвращает только 2, 4, 6 и 8. Функция `stride()` возвращает последовательность значений любого «разлагаемого на шаги» типа, и ее также можно использовать для получения обратного диапазона. Если вы хотите получить обратный диапазон без использования `stride`, используйте функцию `reverse()`, следующим образом:

```
for i in (1...5).reverse() {}
```

Пожалуйста, учтите, что следующий цикл неправилен, и при попытке его выполнить появится сообщение об ошибке:

```
for i in 5...1 {}
```

Как показывает последний пример, `stride()` подходит и для обработки попарно!

## Функции Swift

Функции Swift очень мощные и поддерживают массу инновационных характеристик, включая переменное число параметров [`variadic`], входные-выходные параметры и возможность возвращать кортежи. Переменное число параметров представляет ноль или более параметров заданного типа. Мы проиллюстрируем это в следующем примере (`variadic.swift`):

```
func Sum(integers: Int...) -> Int {
    var total = 0
    for number in integers {
        total += number
    }
    return total
}
var total = Sum(10, 30, 50, -17, 9, 110, -130, 17, 100)
print("The total is \(total)")
```

Функция `Sum` преобразует все переменные целочисленные аргументы в массив констант с именем `integers`, состоящий из элементов `Int`, доступ к которым осуществляется с помощью цикла `for`.

Входные-выходные параметры должны напомнить вам функцию «вызова по ссылке» в C. Они используются, когда вы хотите сохранить изменения, выполненные в функции, по завершении работы этой функции. Входные-выходные параметры Swift иллюстрируются в следующем коде (`inAndOut.swift`):

```
func minMax(inout min: Int, inout max: Int) {
    if min > max {
        let temp = min
        min = max
        max = temp
    }
    var myMin = 100
    var myMax = -10
    print("(Before) Min: \(myMin) and Max: \(myMax)")
    minMax(&myMin, max: &myMax)
    print("(And After) Min: \(myMin) and Max: \(myMax)")
```

Код в файле `inAndOut.swift` дает следующий вывод:

```
(Before) Min: 100 and Max: -10
(And After) Min: -10 and Max: 100
```

Учтите, что Swift требует, чтобы при вызове `minMax()` для второго аргумента указывалась только метка. Поэтому следующие два вызова `minMax()` будут неправильными:



Swift также поддерживает структуры (Structures), классы (Classes), перечисления (Enumerations), протоколы (Protocols), общие конструкции (Generics), обработка ошибок (Error Handling) и функциональное программирование (Functional Programming) и может использоватьсь как скриптовый язык.

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

```
minMax(&myMin, &myMax)
```

```
minMax(min: &myMin, max: &myMax)
```

Сообщение об ошибке, выдаваемое первой строкой кода, будет таким:

```
inAndOut.swift:13:7: error: missing argument label 'max:' in call
```

```
minMax(&myMin, &myMax)
```

```
^
```

```
max:
```

При втором вызове `minMax()` появится следующее сообщение об ошибке:

```
inAndOut.swift:13:7: error: extraneous argument label 'min:' in call
```

```
minMax(min: &myMin, max: &myMax)
```

Способ вызова функции с несколькими аргументами и формируемые сообщения об ошибках могут показаться странноватыми, но сейчас Swift работает именно так.

Кортежи позволяют удобно группировать значения любых типов, а также легко передавать и принимать множество значений без использования структур или отдельных объектов.

```
import Glibc
```

```
func minMax(array: [Int]) -> (min: Int, max: Int) {
```

```
...
```

```
    return (min, max)
```

```
}
```

```
var randomArray = [Int]()
```

```
// Поместить в массив 10 случайных чисел
```

```
for i in 1...10 {
```

```
    var number = random()
```

```
    randomArray.append(number)
```

```
}
```

```
let myMM = minMax(randomArray)
```

```
// Альтернатива
```

```
var (myMin, myMax) = minMax(randomArray)
```

```
print("Максимум \u{20ac}(myMM.max) а минимум \u{20ac}(myMM.min)")
```

```
print("Максимум \u{20ac}(myMax) а минимум \u{20ac}(myMin)")
```

Как видите, есть два способа получения и использования кортежа: в виде одной переменной (`myMM`) или нескольких переменных (`myMin` и `myMax`). Если вам не известно общее количество полей в кортеже, первый способ удобнее. Объявление функции `minMax()` показывает, что она возвращает два значения, а в качестве входных значений требуется массив целых чисел без указания конкретного размера. Основной момент в реализации `minMax()` — это вызов `return` в конце. Команда `import Glibc` позволяет использовать существующие функции C в коде Swift и эквивалентна команде `import Darwin`, которая использовалась в Mac OS X. Полную реализацию функции `minMax()` см. в файле `tuplesRandom.swift`.

Самое мощное управляющее выражение в Swift — это `switch`, так как в его ветвях применимы регулярные выражения и даже сравнение кортежей, чтобы упростить вам жизнь. Это выражение лучше подходит к более сложным условиям и выполняет соответствующий блок кода на основе первого совпадшего соответствия. Так как этому выражению не нужен оператор `break`, то оно не может по ошибке выполнить несколько блоков. Просто

не забудьте, что выражение `switch` должно либо явно покрывать все возможные случаи, либо иметь блок `default` для всех не перекрытых случаев. Вы также получаете совпадшие значения и можете использовать их впоследствии:

```
case (let x, let y): print("You are at \u{20ac}(x) and \u{20ac}(y)")
```

## Необязательные значения в Swift

Необязательные значения [optionals] — еще одна новая функция Swift, которая поможет уменьшить количество ошибок в вашем коде. В следующем коде (файл `optionals.swift`) показано, как они используются со структурой словаря:

```
let Contacts = ["George": "123211212", "Georgia": "23211223"]
let isGeorgiaPresent: String? = Contacts["Georgia"]
if isGeorgiaPresent == nil {
    print("You have no contact named Georgia!")
}
else {
    let contactNumber = isGeorgiaPresent!
    print("The contact number of Georgia is \u{20ac}(contactNumber)")
}
```

Предыдущий код позволяет определить, найден ли соответствующий ключ в словаре, и выполнить соответствующие действия. Знак вопроса (`String?`) означает необязательность значения. Значение `nil` подходит для любого типа и означает «значение отсутствует». Если вы уверены, что у переменной есть значения, то можете принудительно получить значение с помощью восклицательного знака (!), это называется «принудительным развертыванием». Если вы забудете указать '!', вывод для предыдущего кода будет таким:

```
The contact number of Georgia is Optional("23211223")
```

Понятно, такой вывод не слишком полезен. Поэтому помните, что процесс развертывания — который также называется «принудительным извлечением значения» — обязателен.

Теперь реализуем алгоритм пузырьковой сортировки в Swift. Исходный код файла `bubbleSort.swift` таков:

```
func bubbleSort(inout array: [Int]) {
    let elements = array.count - 1
    for var i=elements; i>=1; i-- {
        for var j=0; j<=i-1; j+ {
            // При необходимости рокируем
            if array[j] > array[j+1] {
                let temp = array[j]
                array[j] = array[j+1]
                array[j+1] = temp
            }
        }
    }
}
```

Функция `bubbleSort` использует входной-выходной параметр в качестве входного массива, чтобы изменения в нем сохранились по завершении работы функции. Обратите внимание, что переменная `array.count` содержит общее количество элементов в массиве, а не максимальный номер элемента.

И на этом всё. Продолжайте программировать на Swift, чтобы освоить все его возможности, потому что за этим языком — будущее! 

**Через месяц:  
Проекты  
на Swift**

## Замыкания

Замыкания [Closures] похожи на блоки в Objective-C и на «лямбды» в других языках программирования. Строго говоря, замыкания — небольшие автономные блоки кода, которые можно передавать по вашей программе и в ней использовать. Замыкания очень похожи на объявления функций,

и причина в том, что функции в Swift как раз называются замыканиями!

Важное различие между замыканиями и функциями — в том, что после определения функции ее имя нельзя переназначить, а имя замыкания разрешено использовать повторно, поэтому вы не можете быть

уверенными в том, что одно и то же замыкание делает одно и то же в течение всей программы. Замыкания используются потому, что они компактнее по сравнению с обычными функциями, особенно когда применяются в сочетании с другими функциями, такими как сопоставление [map].



# MongoDB: Для админов

**Михалис Цукалос** описывает всю подноготную администрирования базы данных MongoDB, включая создание набора реплик.

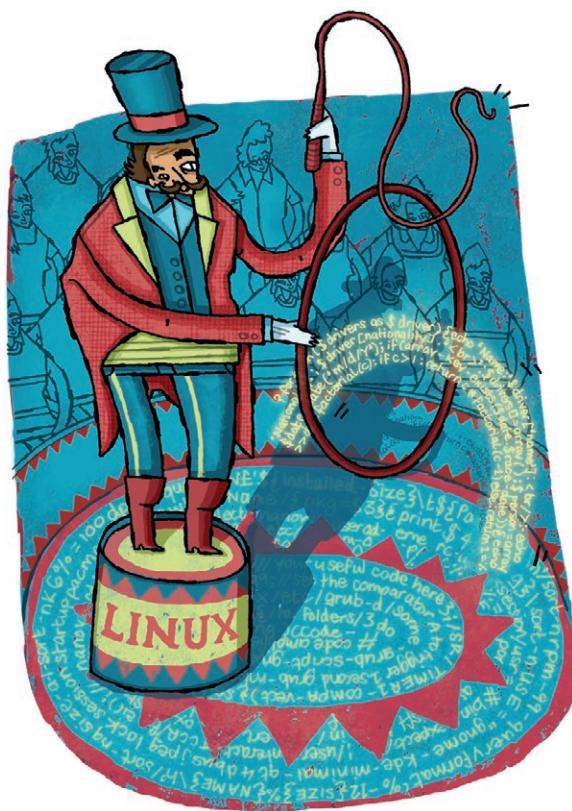


Наш  
эксперт

**Михалис Цукалос** — администратор баз данных, программист, математик и администратор Unix. Обычно ему не платят в монгольских тугриках, но в этом месяце всё будет именно так. Да-да.



Чтобы узнать о MongoDB подробнее, ознакомьтесь с книгами «MongoDB: исчерпывающее руководство», 2-е издание, Кристини Чодоров [Kristina Chodorow] и «MongoDB в действии», 2-е издание, Кайла Банкера [Kyle Banker], Петра Баккама [Peter Bakkm], Шона Верча [Shaun Verch], Дугласа Гэрретта [Douglas Garrett] и Тима Хоукисса [Tim Hawkins].



Давайте поглубже заберемся в MongoDB, ориентированную на документы базу данных NoSQL, которая стала очень популярной. Мы покажем, как выполнять всякие административные задачи на примере таблицы, которую мы создадим и заполним данными, а также посмотрим, как создать набор реплик. С движками баз данных (Storage Engines) мы дела иметь не будем, поскольку команды администрирования баз данных не зависят от используемого движка. Но если вы используете MongoDB на рабочем сервере, вам следует изучить WiredTiger и MMAPv1. Тем более что начиная с версии 3.2 MongoDB, WiredTiger стал движком баз данных по умолчанию.

Информацию по установке свежей версии MongoDB можно найти на сайте [www.mongodb.org/downloads](http://www.mongodb.org/downloads). На момент написания статьи текущая версия — 3.2, и мы установим ее в Ubuntu 14.04 из следующего архива:

```
$ wget https://fastdl.mongodb.org/linux/  
mongodb-linux-x86_64-ubuntu1404-3.2.0.tgz
```

После распаковки запустите такие команды, для запуска сервера MongoDB:

```
$ tar zxvf mongodb-linux-x86_64-ubuntu1404-3.2.0.tgz  
$ cd mongodb-linux-x86_64-ubuntu1404-3.2.0/  
$ cd bin  
$ sudo mkdir -p /data/db  
$ sudo chown "S(id -u -n)" /data/db  
$ sudo ./mongod
```

Последняя команда запустит сервер MongoDB, который сообщит об этом в терминале. Далее мы расскажем, как запустить сервер MongoDB в фоновом режиме без терминала, но если добавить каталог bin MongoDB в свою переменную PATH, это упростит вам жизнь:

```
$ export PATH=/home/mtsouk/  
mongodb-linux-x86_64-ubuntu1404-3.2.0/bin:$PATH"  
Замените /home/mtsouk/mongodb-linux-x86_64-ubuntu1404-3.2.0/  
bin на свой путь.
```

Затем вы сможете определить используемую версию MongoDB — двумя командами:

```
$ ./mongod --version  
db version v3.2.0  
git version: 45d947729a0315accb6d4f15a6b06be6d9c19fe7  
OpenSSL version: OpenSSL 1.0.1f 6 Jan 2014  
allocator: tcmalloc  
modules: none  
build environment:  
distmod: ubuntu1404  
distarch: x86_64  
target_arch: x86_64  
$ ./mongo --version  
MongoDB shell version: 3.2.0
```

mongod — это имя серверного процесса MongoDB, а mongo — имя оболочки MongoDB, которая принимает преимущественно команды JavaScript. Как видите, оба исполняемых файла возвратили одинаковую информацию о версии MongoDB. Большинство задач, представленных в этой статье, выполняются в оболочке Mongo, для запуска которой надо выполнить команду mongo. Если запустить mongo, не указав номер порта, она автоматически попробует подключиться к localhost через порт 27017 — это номер порта по умолчанию, используемый серверным процессом MongoDB. Если вам требуется другой номер порта, запустите mongo с параметром --port, вслед за которым укажите этот номер. (Вы можете подключиться к серверу MongoDB, запущенному на локальном компьютере, через порт по умолчанию.)

Прежде чем пробовать другие команды, запустите следующий код JavaScript из оболочки MongoDB, чтобы добавить какие-то данные в свою базу данных для эксперимента:

```
use LXF  
switched to db LXF
```





```
$ mongod --fork --logpath a.log --smallfiles --oplogSize 50 --port 27101 --dbpath w1 --repSet w --logappend
```

Здесь параметр `--port` задает порт, который будет слушать сервер, `--dbpath` — каталог, где будут содержаться файлы базы данных, `--logpath` указывает на лог-файл, а `--fork` велит ОС запустить процесс в фоне без привязки терминала. Параметр `--repSet` определяет название набора реплик и должен указываться только в том случае, если вы хотите задать набор реплик. Параметр `--logappend` велит процессу *MongoDB* дописывать данные в лог-файл, а не замещать их. Параметр `--oplogSize` определяет максимальный размер файла репликации в МБ и должен указываться только при создании набора реплик.

Попробовав запустить сервер *MongoDB* без соответствующего лог-файла, вы получите следующее сообщение об ошибке:

```
2015-11-29T12:01:54.349+0200 F CONTROL Failed global initialization:2015-12-29T16:02:25.725+0200 F CONTROL [main] Failed global initialization: FileNotOpen Failed to open "/a/path/a.log"
```

## Запросы и индексы

Анализ запроса — исключительно удобный способ понять, почему запрос работает медленно, а также узнать, как выполняется запрос:

```
> db.myData.find({ "x": { $gt: 99990 } }).explain("executionStats")
```

В левой части экранного снимка внизу показан вывод команды `explain("executionStats")` на запрос вверху.

По плану выполнения запроса коллекция сканируется (COLLSCAN), то есть анализируются все документы коллекции. И если в победившем плане есть ключевое слово COLLSCAN, значит, индекс не использовался; а это, в свою очередь, значит, что администратор базы данных должен пересмотреть текущие индексы и задать более подходящие. Правильно заданный индекс способен сильно ускорить выполнение запросов.

Покажем, как определить индекс и выполнить предыдущий запрос. Индекс для ключа "x" создается следующим образом:

```
> db.myData.createIndex({<>x:>1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```



Как всегда, лучший способ изучить *MongoDB* — использовать ее в проекте.

На следующем уроке мы покажем, как создать сайт блога с помощью *MongoDB*, Python и Bottle.

Теперь с `explain("executionStats")` можно выполнить предыдущую команду `find()`. В правой части скриншота внизу показан вывод `explain()` при использовании индекса. Как видите, вывод стал немного другим: новый запрос возвращает результаты путем сканирования ключей индекса (IXSCAN) и поэтому выполняется значительно быстрее предыдущего. Кроме того, запрос теперь анализирует только девять документов, тогда как запрос без индекса анализировал все документы коллекции. Это объяснение более понятно из следующего вывода:

```
"executionTimeMillis" : 0,
"totalKeysExamined" : 9,
"totalDocsExamined" : 9,
```

Вывод показывает, что при выполнении с помощью индекса на выполнение команды `find()` ушло 0 мс, поскольку она, чтобы вернуть девять документов, всего девять документов и проверила (обозначается `totalDocsExamined`).

Функция `getIndexes()` возвращает индексы коллекции, тогда как функция `dropIndex()` удаляет существующий индекс. При этом индекс поля `_id` удалить нельзя.

```
> db.myData.getIndexes()
[ {
  {
    "v" : 1,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "LXF.myData"
  },
  {
    "v" : 1,
    "key" : {
      "x" : 1
    },
    "name" : "x_1",
    "ns" : "LXF.myData"
  }
}]
```

*MongoDB* автоматически создает индекс для поля `_id`, которое есть во всех коллекциях. Все индексы, принадлежащие активной базе данных, можно найти так:

```
db.getCollectionNames().forEach(function(collection) {
  indexes = db[collection].getIndexes();
  print("Indexes for " + collection + ":");
  printjson(indexes);
});
```

Хотя задать индексы просто, надо помнить, что они занимают место на диске, и пользоваться ими только при необходимости.

## Репликация

Репликация по сути означает две вещи, и первая из них — сохранность данных. Изменения будут распространяться и не потеряются, если произойдет что-то плохое. Вторая — доступность данных; это означает, что серверы будут доступны максимальное время; при выходе из строя одного сервера его место без всякого вмешательства пользователя занимает другой.

Большинство наборов реплик имеют три узла: главный [primary], подчиненный [secondary] и арбитр [arbiter]. Арбитр — это специальный узел, который не сохраняет никаких данных и не может стать главным; он просто участвует в голосовании за новый главный узел. Реплики также часто имеют по одному главному узлу и по двум подчиненным. Также набор реплик часто имеет один или

The screenshot shows two terminal windows side-by-side. The left window displays the command `db.myData.find({ "x": { \$gt: 99990 } }).explain("executionStats")` and its output, which includes a detailed plan showing a full collection scan (COLLSCAN). The right window shows the same command but with an index created on the 'x' field, resulting in a much faster execution time (0 ms) and fewer documents examined (9 total keys, 9 total docs). This visual comparison clearly demonstrates the performance benefits of using an appropriate index.

➤ Индекс может существенно сократить время выполнения запроса. Функция `explain(executionStats)` сообщает вам, используется индекс запросом или нет.

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

несколько скрытых узлов, которые не видимы драйверу, не отвечают на запросы пользователей и не могут стать главными. Однако ими можно пользоваться для резервного копирования или аналитики, и они тоже вправе голосовать.

Для демонстрации сценария репликации нужно запустить не менее трех экземпляров *MongoDB*. Хорошо, что все их можно запустить на одном компьютере, хотя это и не лучший вариант для рабочих систем. Итак, запустите три разных процесса, следующим образом:

```
$ mkdir node1; mkdir node2; mkdir arbiter
$ mongod --replSet LXF --dbpath node1 --port 30000 --logappend
--logpath a.log --fork
$ mongod --replSet LXF --dbpath node2 --port 30001 --logappend
--logpath b.log --fork
$ mongod --replSet LXF --dbpath arbiter --port 30002 --logappend
--logpath c.log --fork
```

Затем надо подключиться к компьютеру, который должен выступать в качестве главного узла, и инициализировать набор реплик следующим образом:

```
$ mongo --port 30000
> rs.initiate()
> rs.add("mtsouk-ubuntu:30001")
> rs.add("mtsouk-ubuntu:30002", {arbiterOnly: true})
```

Команда `rs.status()` поможет вам узнать о статусе набора реплик. Если главный узел по какой-то причине выйдет из строя, произойдет следующее: оставшиеся узлы обнаружат сбой, так как не смогут найти отсутствующий узел. Убедившись, что узел действительно недоступен, они проголосуют за новый главный узел. Преуспевший в выборах подчиненный узел станет новым главным и будет принимать операции записи.

Выполнение команды `rs.status()` после остановки главного узла, которую можно выполнить, завершив процесс этого узла командой `kill`, приведет к несколько иному результату по сравнению с приведенным здесь.

## Системные таблицы

*MongoDB* хранит системную информацию в коллекциях, использующих пространство имен `<database>.system.*`, которое *MongoDB* резервирует для внутреннего использования. Поэтому не создавайте коллекций с именами, начинающимися с `system`. В базе данных '`admin`' хранятся многие системные коллекции, в том числе `admin.system.roles`, содержащая пользовательские роли, которые администратор создает и назначает пользователям для выдачи доступа к заданным ресурсам. Имя и пароль для аутентификации каждого пользователя, а также все роли, назначенные пользователю, хранятся в `admin.system.users`, а схема для документов с данными аутентификации — в `admin.system.version`.

В каждой отдельной базе данных есть некоторые коллекции, которые следят за данными, относящимися к этой базе данных. Коллекция `<db>.system.namespaces` содержит информацию о коллекциях базы данных, `<db>.system.indexes` — сведения об индексах базы данных. Однако к коллекции `<db>.system.indexes` надо обращаться не напрямую, а через функцию `getIndexes()`. Коллекция `<db>.system.profile` содержит информацию о профилировании базы данных. Профилирование базы данных поможет определить неэффективные запросы и операции. Наконец, в коллекции `<db>.system.js` хранится специальный код JavaScript для использования в серверном JavaScript.

У каждого экземпляра `mongod` есть собственная «локальная» база данных. В ней хранятся данные, используемые в процессе репликации, и другие данные для конкретного экземпляра. «Локальная» база данных не реплицируется, так как она невидима для процесса репликации.

Системные таблицы *MongoDB* также содержат информацию о файлах GridFS, хранимых в базе данных. Существует две

```
2. mtsouk@mtsouk-ubuntu: ~/mongodb-linux-x86_64-ubuntu1404-3.2.0/bin (ssh)
~$ mongodump
2015-12-29T18:09:37.643+0200      writing LXF.myData to dump/LXF/myData.bson
2015-12-29T18:09:37.644+0200      writing LXF.fs.chunks to dump/LXF/fs.chunks.bson
2015-12-29T18:09:37.645+0200      writing LXF.fs.files to dump/LXF/fs.files.bson
2015-12-29T18:09:37.664+0200      done dumping LXF.fs.files (3 documents)
2015-12-29T18:09:38.168+0200      done dumping LXF.fs.chunks (138 documents)
2015-12-29T18:09:38.239+0200      done dumping LXF.myData (100000 documents)
~$ ll dump/
total 12
drwxrwxr-x  3 mtsouk mtsouk 4096 Dec 29 18:09 .
drwxr-xr-x 25 mtsouk mtsouk 4096 Dec 29 18:09 ..
drwxrwxr-x  2 mtsouk mtsouk 4096 Dec 29 18:09 LXF/
~$ ll dump/LXF/
total 39004
drwxrwxr-x  2 mtsouk mtsouk 4096 Dec 29 18:09 ..
drwxrwxr-x  3 mtsouk mtsouk 4096 Dec 29 18:09 ../
-rw-rw-r--  1 mtsouk mtsouk 35510003 Dec 29 18:09 fs.chunks.bson
-rw-rw-r--  1 mtsouk mtsouk 165 Dec 29 18:09 fs.chunks.metadata.json
-rw-rw-r--  1 mtsouk mtsouk 425 Dec 29 18:09 fs.files.bson
-rw-rw-r--  1 mtsouk mtsouk 84 Dec 29 18:09 fs.files.metadata.json
-rw-rw-r--  1 mtsouk mtsouk 4400000 Dec 29 18:09 myData.bson
-rw-rw-r--  1 mtsouk mtsouk 137 Dec 29 18:09 myData.metadata.json
~$ mongorestore --quiet
e ■
```

➤ Утилиты `mongodump` и `mongorestore` помогут выполнить резервное копирование и восстановление базы данных *MongoDB*.

системные таблицы: одна для имен файлов, а другая — для различных фрагментов с фактическими двоичными данными файла. По умолчанию GridFS использует две коллекции с именами с префиксом `fs`: `fs.chunks` и `fs.files`.

Желая найти все файлы GridFS, которые хранятся в базе данных, вы можете опросить системную таблицу, содержащую эту информацию. Далее приведен вывод запроса к таблице `fs.files`:

```
use LXF
switched to db LXF
> db.fs.files.find()
{ "_id" : ObjectId("5682af61bd65690bcc000001"), "chunkSize" :
261120, "uploadDate" : ISODate("2015-12-29T16:05:53.642Z"),
"length" : 28549, "md5" : "1494011900cf195d44165191d042a124",
"filename" : "/home/mtsouk/aFile.zip" }
{ "_id" : ObjectId("5682af7cbd65690bdb000001"), "chunkSize" :
261120, "uploadDate" : ISODate("2015-12-29T16:06:20.882Z"),
"length" : 28549, "md5" : "1494011900cf195d44165191d042a124",
"filename" : "aBinaryFile.zip" }
{ "_id" : ObjectId("5682af85bd65690be8000001"), "chunkSize" :
261120, "uploadDate" : ISODate("2015-12-29T16:06:29.642Z"),
"length" : 35444349, "md5" : "92c7ae236c2ea6326a603c-
9cffb9437b", "filename" : "mongosniff" }
```

Или можно воспользоваться утилитой `mongofiles` для вывода всех файлов GridFS, присутствующих в базе данных:

```
$ mongofiles -d LXF list
2015-12-29T18:07:54.895+0200 connected to: localhost
/home/mtsouk/aFile.zip 28549
aBinaryFile.zip 28549
mongosniff 35444349
```

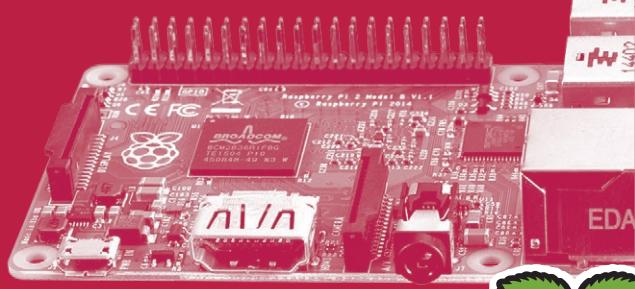
Если вам нужна дополнительная информация о `mongofiles`, зайдите на <http://bit.ly/MongofilesDocs>.

## Резервирование и восстановление

При работе с базой данных обязательно нужна стратегия резервного копирования, так как все системы в конце концов выходят из строя. Хотя шардирование и репликация *MongoDB* снижают потребность в резервном копировании, что-то плохое может произойти всегда. А уж если у вас всего один серверный процесс, делать резервные копии обязательно. Две основные

утилиты для этих целей — `mongodump` и `mongorestore`, и они помогут вам сделать резервную копию базы данных *MongoDB* и восстановить ее соответственно.

На рисунке вверху показано использование `mongodump` и `mongorestore`. Команда `mongorestore` была запущена с параметром `--quiet`, чтобы скрыть все служебные сообщения.



Ваша порция смачных новостей, обзоров и учебников от Raspberry Pi

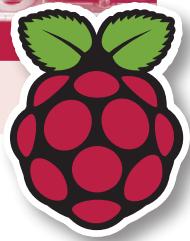


Привет...

**K**ак-то раз у себя дома, на юге Франции, я проснулся очень рано. Решил включить шоу *BBC Click* и увидел сюжет Эл Джей Рича [LJ Rich]: «„Сделай сам“ в технологии — реально ли это?» Там говорилось о движении Maker, ИТ-мастерских, Arduino и — как же без него — «улетном успехе Raspberry Pi». Успехе... чего? Так для меня все и началось, и сегодня я очень даже «в теме» насчет того, что происходит здесь, во Франции, хотя я точно не профи, просто большой фанат Raspberry.

Rhône Valley Raspberry Jam мы с друзьями проводим с февраля прошлого года, и только что завершилось наше шестое мероприятие. Также я активноучаствую в развитии Code Club во Франции (и других инициатив в области кодинга). Raspberry Pi дал мне шанс путешествовать и говорить о «цифровом образовании» плюс массы новых знакомств.

По части применения Pi и численности сообщества Франция сильно отстает от других стран. Мы надеемся, что в 2016 г. это изменится. Самое сложное здесь — преодолеть языковой барьер. Даже самые «франкофонные» фанаты зачастую решают подтянуть английский, благодаря этому «нанокомпьютеру». Для меня главной фигурой Pi во Франции является Франсуа Мок [Francois Mocq], преподаватель информатики, в чьем блоге ([www.framboise314.fr](http://www.framboise314.fr)) уже более 600 постов. Он уже стал одним из самых продаваемых авторов в сфере ИТ и сейчас помогает мне переводить учебные материалы Raspberry Pi Foundation, чтобы они стали доступны французским поклонникам Pi.



## Pi для ребенка...

Пожертвуйте свой неиспользуемый Pi детям из Мали, в западной Африке.

**R**aspberry Pi создавался как потрясающий инструмент обучения; и, безусловно, стал таким. На четвертом году своего существования Raspberry Pi открывает новые горизонты образования — одним из самых дальних стал Мали, в Западной Африке, где французский благотворительный проект Malinux Télé (<http://malinustele.tuxfamily.org>), используя Pi, обучает малийских детей.

С помощью собственной ветки Raspbian они подключают старые (и новые) модели Pi к ЭЛТ-экранам, работающим от солнечных батарей, чтобы учить детей программированию. Чтобы принять участие, узнайте подробности на <http://bit.ly/MalinuxTelebelow> или отправьте свои Pi по адресу: Association Bilou Toguna, 25 Rue Henri Martin, 31700 Blagnac, Франция.

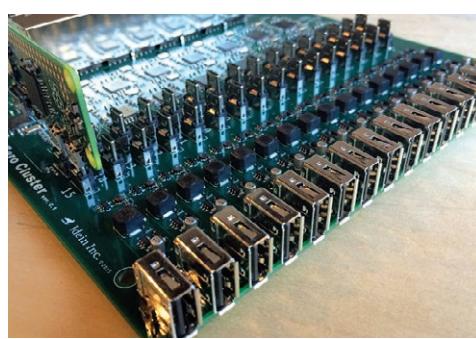


► Малийские дети учатся на ваших старых Pi.

## 1 × 16 = Pi!

Кластер Multi-Pi — штука интересная.

**Я**понская компания под названием Idein разработала плату мульти-Pi, способную подключать и взаимодействовать с 16-ю Pi Zero. Есть вероятность, что у этого продукта, задуманного изначально как основа для научных проектов на базе GPU, будет и коммерческий релиз. Не пропустите! <http://bit.ly/LXF208cluster>



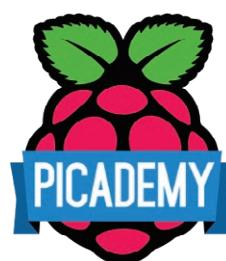
► Pi — это сила!

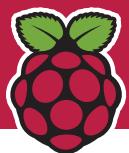
## Учите Pi

Состоялись мартовские курсы.

**П**едагоги! На зависть своим мастеровым друзьям и коллегам, записывайтесь на Picademy, проводимую @Google. В Манчестере состоялся бесплатный, двухдневный Непрерывный курс повышения квалификации [Continued Professional Development] для профессиональных преподавателей.

Занятия проходили в новом здании Digital Garage Google 1-2 марта 2016, 14-15 марта 2016 и 22-23 марта 2016. Подробности — на <http://bit.ly/PicademyGoogleManchester>





# Explorer pHAT

**Лес Паундер** тестирует свежую версию многофункциональной платформы для нового Pi Zero, которую вполне можно купить на карманные деньги.

## Вкратце

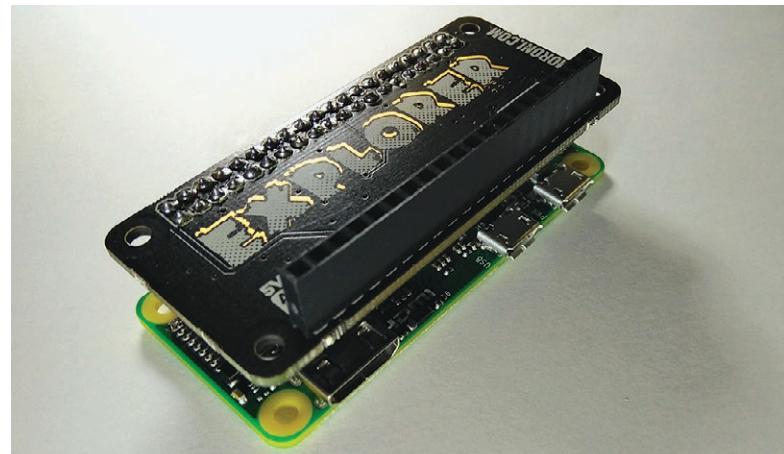
» Pimoroni создал очередную плату-приставку в серии плат, нацеленных на мыслителей и самоделкинх. Разработанная для нового Pi Zero, но совместимая со всеми недавними Raspberry Pi, Explorer pHAT предлагает ряд защищенных контактов входов и выходов, позволяющих экспериментировать без вреда для вашего Pi. pHAT также обеспечивает управление мотором для движущихся проектов.

Raspberry Pi Zero только появился, а огромное сообщество уже создало платы-приставки к компьютеру-крошке. Pimoroni создал новую версию своей популярной платы Explorer HAT, расширив свой диапазон до трех плат. Производство стартовало в начале 2014 г. с двух плат: Explorer HAT, которая шла с емкостным сенсорным входом, светодиодами и набором входов и выходов. Но в стандартной версии не хватало контроллера двигателя, и он был добавлен во вторую, расширенную версию — Pro. А в последней версии Pimoroni мы видим усовершенствования.

Explorer pHAT отлично встает поверх 40-контактного GPIO из Pi Zero и совместим со всеми 40-контактными GPIO Raspberry Pi. Плата требует небольшой пайки — собственно, как и сам Pi Zero. Explorer pHAT — урезанная версия Explorer Pro, где удалены емкостные входы и контакты I2C/3 В, но оставшиеся элементы весьма впечатляют.

Explorer pHAT идет с двумя Н-драйверами мотора, способными менять полярность мотора, позволяя ему работать в двух направлениях, чтобы робот мог двигаться в любом направлении с большой точностью. Выходы двигателя предлагают программно управляемый метод изменения их мощности, так что скорость ваших проектов станет варьируемой. Explorer pHAT также имеет четыре аналоговых входа (для Pi это не стандартно, но представлено на платах Arduino), позволяющие использовать в проекте аналоговые сенсоры, такие как DHT 11/22.

Плата также идет с набором 5-В входов и выходов для использования с другими



» Explorer pHAT отлично встает поверх Pi Zero, но он весьма комфортабельно чувствует себя и на других моделях 40-контактных GPIO Pi.

компонентами. Управление светодиодом относительно простое и требует использования резистора, чтобы светодиод не перегорел. Вместо управления током, идущим к светодиоду, контролируется заземление, являющееся выходным контактом платы. Каждый выход платы может обеспечить максимум 500 мА через массив Дарлингтона [Darlington array], высоковольтный переключатель. От контакта получается достаточно большой ток, но максимальный выход с объединенных контактов ограничен 1 А. Вход работает с 5-В логическим устройством, но может срабатывать от любого напряжения от 2 В до 5 В, что очень удобно при работе с ультразвуковыми сенсорами и пассивным инфракрасным датчиком (PIR).

присутствующие в версии Pro, и светодиодов, представленных в обеих предыдущих версиях, но цена и наличие контроллеров двигателя более чем компенсируют эти упущения. С помощью Pi Zero и Explorer pHAT легко создать моторизированный проект, вставляемый в самые маленькие корпуса.

Цена платы и ее возможности исключительны и отлично вписывается в новую экосистему Zero. Мы склонны согласиться с Эбеном Аптоном [Eben Upton], основателем Raspberry Pi Foundation, по поводу его недавнего высказывания о приставках к Zero: «Я полагаю, что приставки стоимостью от £10 до £15 будут иметь спрос на рынке, поскольку люди захотят купить аксессуары для этих устройств», и Explorer pHAT — один из них. [LXF](#)

## Исключительные функции

Библиотека Python для Explorer pHAT весьма надежна и охватывает версии 2 и 3. Используя очень простой для понимания синтаксис, можно управлять состоянием множества входов и выходов на плате. Режимы — такие как плавное изменение уровня сигнала, мигание и переключатели — обрабатываются за вас и упрощают новичкам процесс привыкания.

Explorer pHAT совместима со всеми недавними Pi и отлично вписывается в серию плат Explorer. Да, нам будет не хватать емкостных входов в стиле MaKey MaKey — приставка-тачпад к предметам повседневного пользования, — прим. пер.,



## Свойства навскидку



### Мала, но мощная

У платы четыре выхода, по 500 мА на каждый, и специализированные Н-контроллеры, пригодные для 5-В моторов.

### Простой язык

Прилагаемые библиотеки Python удивительно интуитивны, и в кратчайшие сроки вы будете создавать свои проекты.

## LINUX FORMAT Вердикт

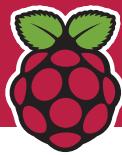
### Explorer pHAT

Разработчик: Pimoroni  
Сайт: <https://shop.pimoroni.com>  
Цена: £ 10

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

» Снабжает нужной функциональностью хакеров аппаратуры, а библиотека Python превосходна.

Рейтинг **8/10**



# Ruby: Сочиним случайную музыку

Лес Паундер рассматривает, как немного изучить Ruby, программируя Raspberry Pi и сочиняя фрагмент случайной музыки с *Sonic Pi*.



## Наш эксперт

**Лес Паундер** — участник образовательной команды Raspberry Pi Foundation и путешествует по стране, заставляя Pi выделять всякие штуки. Он пишет блог на <http://bigles.net>.



**K**огда мы думаем о языках для общения с Raspberry Pi, мы, естественно, подразумеваем Scratch и Python. Однако *Sonic Pi* — программируемый в реальном времени синтезатор — использует Ruby, популярный язык, созданный Юкихиро “Matz” Мацумото и тоже достойный изучения.

На этом уроке мы изучим ряд приемов кодирования на Ruby, используя приложение *Sonic Pi*. Мы будем использовать переменные для хранения данных; циклы для повторения проекта; массивы для хранения наборов нот и условные операторы для изменения музыки после тестирования. К концу этого проекта вы создадите нечто уникальное; это не музыка в привычном понимании, но первый шаг в автоматическом ее создании с помощью генерации случайных чисел.

Для этого проекта годится любая модель Raspberry Pi; вам потребуется аудио на телевизоре (или комплект 3,5-мм динамиков) и последняя версия Raspbian Jessie. Весь код проекта — на <http://bit.ly/LXF208Randomness>.

Начнем мы с загрузки Pi в рабочий стол, затем откроем приложение *Sonic Pi*, которое можно найти в меню Programming. Через несколько секунд *Sonic Pi* будет готов к работе. *Sonic Pi* использует несколько буферов для хранения кода; в буфер 0 мы начнем писать на Ruby.

Первая строка кода играет ноту, *Sonic Pi* использует MIDI (цифровой интерфейс музыкальных инструментов) для представления нот числами, так что для ноты «до» мы используем 60: play 60.

Затем нажмите кнопку Run. Если ничего не слышно, нажмите на меню Prefs и измените аудиовыход в соответствии с вашим устройством. Вы должны услышать звук «бом».



▶ *Sonic Pi* имеет очень понятный интерфейс, что способствует его освоению. Ему вторит и исключительно чистый и легко читаемый в редакторе код Ruby.

Далее используем два понятия кодирования для этой строки: цикл повтора ноты и переменную:

```
pace = 1
loop do
  play 60
  sleep pace
end
```

Переменная `pace` [англ. темп] используется для хранения числового значения — целого или с плавающей точкой. Используя этот метод, можно создать место для хранения значения темпа проекта, вместо того, чтобы жестко закодировать его для всего проекта. Цикл, как видите, начинается с `do` и закрывается с `end`. Важно помнить, что каждый цикл должен быть правильно открыт и закрыт. В данном примере цикл выполняется бесконечно.

## Темп и потоки

Чтобы изменить темп проекта, можно просто изменить значение темпа, например, если он меньше 1, использование числа с плавающей точкой, такого как 0.3, сделает это очень быстро. В примере ниже я добавил еще две строки после строки четыре, но перед концом цикла:

```
play 70
sleep pace
```

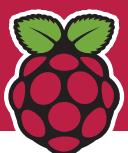
Далее мы добавим некоторый код, чтобы содержать наш цикл в потоке. В Ruby легко оперировать с потоками, так что мы можем запустить код в несколько потоков. В *Sonic Pi* мы используем это для создания слоев создаваемой музыки.

Наш код теперь будет выглядеть так (помните, что начав цикл с `do`, мы должны завершить его с `end`, тот же принцип применяется и к потокам):

```
pace = 0.3
in_thread do
  loop do
    play 60
    sleep pace
    play 70
    sleep pace
  end
end
```

Теперь создадим еще один поток, с другим набором нот. В этом случае мы скопируем и вставим исходный поток и понизим высоту звука.

```
in_thread do
  loop do
    play 50
    sleep pace
```



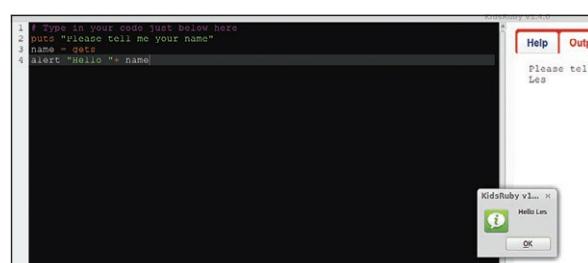
## Sonic Pi и Minecraft

*Sonic Pi* — отличный способ познакомить детей с Ruby, но, как мы знаем, многие дети подсели на *Minecraft* и тратят много времени на игру. А знаете ли вы, что *Sonic Pi* также можно использовать для управления *Minecraft* подобно библиотеке Python?

Для этого потребуется открыть *Minecraft* и *Sonic Pi*. Это возможно на всех Raspberry Pi, но лучшие результаты получаются на Pi 2. В *Sonic Pi* мы можем интегрировать команды *Minecraft* в нашу

композицию — например, телепортировать Стива два блока по воздуху, и это будет происходить в такт музыке — добавив `mc_teleport(0,2,0)`.

Благодаря Джеймсу Робинсону [James Robinson] из Raspberry Pi Foundation, *Sonic Pi* можно настроить реагировать на ввод в мире *Minecraft*, так как он написал клавиатурную игру, где игрок сочиняет музыку с помощью клавиатуры *Minecraft*. *Sonic Pi* имеет раздел помощи, который охватывает функциональность *Minecraft*.



» **Kids Ruby** — отличное приложение и поможет любому сделать первые шаги с Ruby.

```
play 55
sleep pace
end
end
```

Итак, теперь у нас одновременно есть два потока, создающих ритм. Для следующего эксперимента используем случайное число, чтобы изменить высоту ноты. С помощью `rand_i(70)` выберем случайное число между 0 и 70. Также используем один из многих инструментов, доступных в *Sonic Pi*, известный как синтезатор, чтобы менять звучание нашей музыки. В данном случае используем `dsaw` — цифровую пилю/гудящий шум. Поэтому для второго потока наш код теперь выглядит так:

```
in_thread do
loop do
use_synth :dsaw
play 50
sleep pace
play rand_i(70)
sleep pace
end
end
```

Создастся едва ощутимое различие в мелодиях при каждом воспроизведении музыки.

Давайте добавим еще поток в наш проект, что добавит слой в создаваемую нами какофонию звуков. Этот поток использует другой цикл и играет любые ноты с помощью синтезатора 'blade', который напоминает работу Вангелиса над саундтреком к фильму *Бегущий по лезвию*. На сей раз воспроизводить ноты будем другим методом. Прежде мы просили *Sonic Pi* играть ноту, потом «уснуть», и довольно быстро это становится громоздким. Поэтому используем функцию `play_pattern_timed`, которая принимает два условия: массив для хранения нот, которые мы хотим играть, и управление временем нот — в данном случае мы используем переменную `pace`, созданную ранее. Ваш новый поток должен выглядеть следующим образом:

```
in_thread do
loop do
use_synth :blade
play_pattern_timed [70,72,74],[pace]
end
end
```

### Играть и слушать музыку

И последний, самый амбициозный поток: мы используем цикл, чтобы постоянно проверять значение переменной в серии условий

`if..elsif..else`. Если значение переменной удовлетворяет условиям, то код, содержащийся в нем, запускается:

```
in_thread do
loop do
with_fx :reverb do
random = rand_i(100)
if random > 75 then
use_synth :dull_bell
play random
sleep 0.2
elsif random > 50 and random < 75 then
use_synth :pulse
play random
sleep 0.2
else
use_synth :piano
play random
sleep 0.2
end
end
end
end
```

Здесь мы создали новую переменную с именем `random` и использовали ее для хранения случайного целого числа от 0 до 100. Затем применяем условные операторы для проверки значения переменной `random`, и если она больше 75, то инструмент меняется на `dull_bell` и играется музыкальная нота. Цикл повторяется, и случайное число сравнивается снова. Если оно не проходит две проверки, то включается условие `else` и для проигрывания ноты выбирается фортепианный синтезатор.

Как-то так. С помощью программной логики, нескольких случайных чисел и *Sonic Pi* мы создали мелодию, которая была написана компьютером с использованием случайных чисел. [\[EXP\]](#)

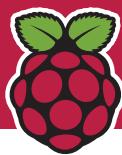
### Скорая помощь

Ruby — отличный язык и является прекрасным введением в типизированные языки. Если вам понадобится помочь по *Sonic Pi*, музыкальной или вычислительной логике, то *Sonic Pi* имеет отличную справочную систему, расположенную в нижней левой части экрана.



» У *Sonic Pi* прекрасная справочная система, которая включает и кодирование, и музыкальную композицию проекта. Взгляните для вдохновения, если застрянете.

» **Подпишитесь на печатную или электронную версию** на [www.linuxformat.ru/subscribe](http://www.linuxformat.ru/subscribe)!



# OwnCloud 8.0: Новее и лучше

**Маянк Шарма** показывает, как взять под контроль хранение своих данных и распределенный доступ к ним в свежем релизе облака.



Наш  
эксперт

Маянк Шарма довольно часто витает в облаках, так что было вполне логично попросить его написать про его приключения в мире облачных технологий.



СОВМЕСТИМО  
С ПК И РП



Чтобы просмотреть статистику кэша APC, скопируйте его скрипт в ваш DocumentRoot с помощью cp /usr/share/doc/phpapc/apc.php /var/www/ и затем просмотрите статистику в web-браузере на <http://localhost/apc.php>.

**Н**ужно ли вам удобство вездесущего Dropbox-подобного сервиса хранения без горечи расставания с крупной наличностью и предоставления ваших данных третьей стороне? OwnCloud — одна из лучших программ с открытым кодом, помогающая создать свой собственный частный и защищенный облачный сервис. С помощью OwnCloud можно синхронизировать и делиться своими личными данными и получать к ним доступ с любого устройства, подключенного к Интернету. Для большей безопасности OwnCloud также может шифровать ваши файлы. ПО умеет управлять файлами различных форматов, и можно улучшить ее практичность, добавив ряд других приложений.

Как и с другими облачными онлайн-сервисами, с OwnCloud можно синхронизировать файлы, используя web-браузер или настольный клиент в Windows, Mac и Linux, а также мобильные клиенты для устройств Android и iOS. Более того, ваш сервер OwnCloud хранит старые версии всех измененных файлов и позволяет вернуться к старой версии без особых усилий.

В данном руководстве мы настроим сервер OwnCloud на дистрибутиве Raspbian для Raspberry Pi. ПО сервера имеет весьма скромные требования и в небольших и контролируемых средах — таких, как ваш дом — хорошо работает даже на Raspberry Pi Model B. Вам также понадобится USB-диск для хранения данных, и для максимальной надежности и производительности лучше всего использовать диск с автономным питанием, не вытягивающий энергию из Raspberry Pi. Прежде чем начать настраивать сервер, убедитесь, что у Raspberry Pi есть статический IP-адрес. Простейший способ сделать это — привязать IP-адрес к уникальному MAC-адресу вашего Pi на странице администрирования вашего роутера. Мы примем адрес нашего Pi за 192.168.3.111, но ваш будет другим!

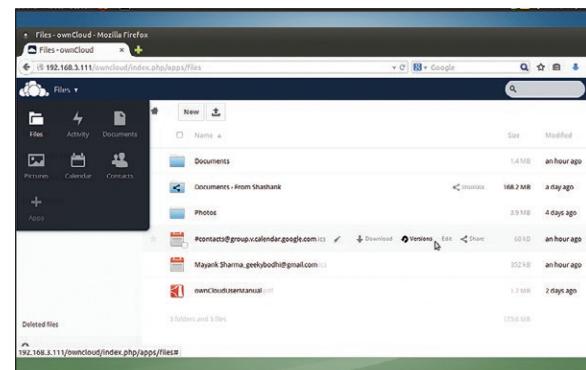
## Установка OwnCloud

Поскольку Raspbian основан на Debian, можно загрузить пакеты OwnCloud из репозитория Debian. Запустите терминал и добавьте репозиторий OwnCloud с помощью

```
$ echo 'deb http://download.opensuse.org/repositories/isv:/ownCloud:community/Debian_7.0/' | sudo tee /etc/apt/sources.list.d/owncloud.list
$ wget http://download.opensuse.org/repositories/isv:OwnCloud:community/Debian_7.0/Release.key
$ sudo apt-key add - < Release.key
```

Репозитории можно обновить командой \$ sudo apt-get update. Затем установите сервер OwnCloud и все требуемые зависимости с помощью \$ sudo apt-get install owncloud. Это также загрузит и настроит базу данных MySQL, и вас попросят задать пароль root.

В добавок к установке требуемых компонентов, вышеупомянутая команда автоматически настроит web-сервер Apache, чтобы взаимодействовать с установкой OwnCloud. Для корректной работы OwnCloud понадобится включить определенные модули Apache. Введите в терминале



У OwnCloud хорошо продуманная панель управления, основанная на браузере — при множестве функций, она легка в управлении.

```
$ sudo a2enmod headers rewrite env
```

Затем перезапустите Apache с помощью \$ sudo apachectl restart.

Если вы хотите загружать файлы размером более 2 МБ, вам придется настроить файл конфигурации PHP. Для этого откройте в текстовом редакторе файл конфигурации PHP, **php.ini**, находящийся в **/etc/php5/apache2**. Найдите переменные **upload\_max\_filesize** и **post\_max\_size** и измените их значения с 2М на что-нибудь вроде 1024М или даже 2G.

Опционально, в более крупных установках также можно установить акселератор PHP — APC, чтобы ускорить установку OwnCloud. Загрузите компоненты с помощью sudo apt-get install php-apc, затем откройте файл настройки APC и добавьте

```
$ sudo mkdir /etc/php5/conf.d
$ sudo nano /etc/php5/conf.d/20-apc.ini
extension=apc.so
apc.enabled=1
apc.shm_size=12M
```

Затем включите кэширование, перезагрузив Apache с помощью \$ sudo apachectl restart

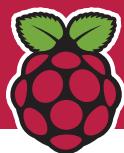
Итак, сервер настроен и сконфигурирован; пора подготовить средство хранения. Подключите USB-диск в Pi и введите в терминале sudo blkid. Вероятно, USB-диск смонтирован как /dev/sda1, если у вас не прикреплено других USB-дисков. Обратите внимание на соответствующий UUID, который выглядит наподобие 6154-F660. Теперь создайте папку для монтирования этого диска, командой

```
$ sudo mkdir /media/owncloud
```

Затем смонтируйте диск, используя

```
$ sudo mount -t vfat -o umask=007,auto(uid=33,gid=33) /dev/sda1 /media/owncloud
```

Эта команда предполагает, что у вашего диска файловая система FAT32 и он монтируется на /dev/sda1. Когда диск корректно смонтируется, вы сможете отредактировать файл **fstab**, чтобы обеспечить ему автоматическое монтирование:



```
$ sudo nano /etc/fstab
UUID=6154-F660 /media/owncloud/vfat
rw,umask=007,auto,uid=33,gid=33 0 0
```

## Настройте облако

Всё это было для установки компонентов сервера. Теперь вам предстоит настроить свое облако. Запустите web-браузер и перейдите к экземпляру установки OwnCloud на 192.168.3.111/owncloud. Поскольку это новая установка, вас попросят создать учетную запись пользователя для администратора OwnCloud.

Затем надо будет попросить OwnCloud использовать базу данных MySQL и хранить файлы на монтированном USB-диске. Для этого нажмите на выпадающее меню Storage & Database [Хранилище и база данных], введите /media/owncloud/data в текстовом окне, соответствующем пункту Data Folder [Папка с данными], и выберите опцию MySQL/MariaDB в разделе Database. Вас попросят ввести подробности подключения сервера базы данных, так что просто введите localhost в качестве хоста и root в качестве имени пользователя вместе с паролем, который вы задали, когда база данных загружалась вместе с OwnCloud.

Ну вот — вы настроили OwnCloud. Теперь можете войти в свой облачный сервер как администратор с помощью аутентификационной информации, которую вы установили для OwnCloud. Начните использовать сервер для загрузки и скачивания файлов можно прямо сейчас, но давайте уделим некоторое время на приборку. Для начинающих, когда вы входите в сервер OwnCloud, нажмите на выпадающее меню рядом с вашим именем пользователя и нажмите на Personal [Личные настройки]. Здесь вы можете изменить настройки своей учетной записи, такие как пароль и отображаемое имя. Также, если ваше облако будет использоваться несколькими людьми, советуем распределить пользователей по разным группам. Чтобы сделать это, выберите опцию Users [Пользователи] в выпадающем меню. При добавлении пользователей вы можете ограничить предоставляемое им пространство для хранения, а то и поделиться своими правами администратора с другими пользователями.

Теперь вы готовы загрузить данные на ваш сервер OwnCloud. Войдя, вы окажетесь в разделе Files [Файлы]. Чтобы загрузить файл, нажмите на плюс (+) и выберите Upload [Отгрузить] в выпадающем меню. Чтобы распределить файлы по папкам, нажмите на плюс и выберите опцию Folder [Папка].

Если вы загрузили файл в понятном для OwnCloud формате, можно щелкнуть по его имени, чтобы просмотреть и отредактировать

его. OwnCloud умеет визуально отображать хранимые им данные в разных видах: например, нажмите на выпадающее меню Files в левом верхнем углу интерфейса и выберите опцию Pictures [Изображения]. Это отображение поможет вам просматривать изображения в вашем облаке, отфильтровывая все остальные типы содержимого.

Иной способ загрузить файлы на сервер — использовать протокол WebDAV, с которым вы можете получить доступ к своему облачному серверу из менеджера файлов. Например, в менеджере файлов Files нажмите Ctrl+I, чтобы включить область размещения. Здесь вы можете указать на свой сервер OwnCloud, например, dav://192.168.3.111/owncloud/remote.php/webdav. После аутентификации хранилище OwnCloud смонтируется, и вы сможете взаимодействовать с ним, как с обычной папкой.

Чтобы поделиться загруженными файлами, перейдите в раздел Files в web-интерфейсе и нажмите кнопку Share [Поделиться] слева от имени файла, что позволит вам выбрать, с какими пользователями или группами вы хотите поделиться и собираетесь ли вы предоставить им право редактировать и удалять файлы. Вы также можете поделиться с кем-нибудь не зарегистрированным на вашем сервере OwnCloud. Нажмите на Share with Link [Поделиться ссылкой], и OwnCloud отобразит ссылку на элемент, которым вы сможете поделиться с каждым в Интернете. Вы также можете защитить ссылку паролем и установить срок действия этого пароля — дату его истечения.

Хотя вы можете взаимодействовать с облаком через web-интерфейс, куда проще использовать один из его официальных клиентов. У OwnCloud есть клиенты для всех основных рабочих столов и мобильных платформ. Эти клиенты также помогают вам с легкостью синхронизировать папки с вашего рабочего стола с вашим сервером OwnCloud.

## Настройте клиенты

Большинство настольных дистрибутивов размещают в своих официальных репозиториях клиент Linux. Можно также получить последнюю версию клиента, добавив соответствующий репозиторий для вашего дистрибутива с <http://bit.ly/OwnCloudClients>.

Страница содержит инструкции для популярных дистрибутивов, включая Debian, Fedora, Ubuntu, openSUSE и т. д. Чтобы скачать клиенты для других платформ, перейдите на страницу Downloads на сайте OwnCloud (<https://owncloud.org/install/#install-clients>). Мобильные клиенты лучше всего скачивать с Apple App Store или Google Play Store.



Если у вашего диска есть раздел NTFS, установите драйвер NTFS с помощью sudo apt-get install ntfs-3g и используйте -t ntfs-3g в команде mount.

## Готовые решения

Хотя установить и настроить сервер OwnCloud с нуля не так уж сложно, существует пара способов сэкономить время и силы. Парни из PetRockBlog (<http://blog.petrockblock.com>) написали скрипт, автоматизирующий весь процесс установки. Скрипт скачивает и настраивает установку OwnCloud в дистрибутиве Raspbian. Однако, в отличие от нашего урока, скрипт вместо web-сервера Apache использует web-сервер Nginx.

Чтобы использовать скрипт, установите требуемые компоненты с помощью

```
$ sudo apt-get install git dialog
```

Затем установите скрипт с помощью

```
$ git clone git://github.com/petrockblog/OwncloudPie.git
```

Создается папка с именем OwncloudPie. Перейдите в эту папку по \$ cd OwncloudPie и сделайте скрипт исполняемым, используя

```
$ chmod +x owncloudpie_setup.sh
```

Затем выполните его:

```
$ sudo ./owncloudpie_setup.sh
```

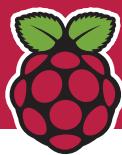
Теперь вам нужно лишь дать ему немного времени для скачивания всех компонентов и настройки сервера для вас. Установив OwnCloud из скрипта, можете запускать его снова, чтобы обновить установку, как только появится новая версия OwnCloud.

Если вы принадлежите к типу с более авантюрными наклонностями, попробуйте установить

на свой Raspberry Pi дистрибутив arkOS (<https://arkos.io>). Вдобавок к OwnCloud, этот дистрибутив имеет другие приложения, которые вы можете использовать, чтобы и дальше контролировать свои данные. Перейдите на страницу Downloads [Загрузки], затем скачайте и распакуйте установщик для Raspberry Pi. После этого вставьте SD-карту и запустите установщик arkOS командой \$ sudo ./arkos-install.

Следуйте шагам установщика для скачивания образа из Интернета и его установки на вашу SD-карту. После этого вам надо загрузить свой Raspberry Pi с карты и перейти на <http://arkos:8000>, чтобы настроить свой сервер.

**» Подпишитесь на печатную или электронную версии** на [www.linuxformat.ru/subscribe](http://www.linuxformat.ru/subscribe)!



## Синхронизируйте папки и делитесь ими

Когда клиент установится, он предложит вам ввести ваши аутентификационные данные, чтобы подключиться к *OwnCloud*. После подключения клиенты Linux создают локальную синхронизируемую папку с именем **owncloud** в папке **home**, например, **/home/bodhi/owncloud**. Любые файлы, перемещенные в эту папку, будут автоматически синхронизированы с сервером. Можно также указать одну или несколько папок на локальном компьютере для синхронизации с сервером *OwnCloud*. Если папка имеет распределенный доступ с несколькими пользователями, изменение файла на одном компьютере, автоматически распространится на другие.

При взаимодействии с другими пользователями вы оцените систему управления версиями *OwnCloud*, которая создает резервные копии файлов перед их изменением. Эти резервные копии доступны из выпадающей опции **Version** [Версия], соответствующей каждому файлу вместе с кнопкой **Restore** [Восстановить] для возврата к более старой версии.

Кроме файлов, с вашим сервером *OwnCloud* можно синхронизировать ваш календарь и адресную книгу. Следуйте руководству (см. стр. 95 справа), чтобы включить приложения *Calendar* и *Contacts*. После включения обеих программы верхнее левое выпадающее меню обзаведется опцией **Calendar** [Календарь] и **Contacts** [Контакты].

Затем импортируйте в свой облачный сервер контакты и календарь из ваших прежних приложений. *OwnCloud* поддерживает популярный формат файла vCard (с расширением файла VCF) и почти все популярные приложения электронной почты, включая онлайновые, такие как Gmail, если вы экспортируете их адресные книги в этот формат. Похожим образом, календарь можно

импортировать в популярный формат iCAL. Прежде чем идти дальше, позаботьтесь скачать VCF- и iCAL-файлы из ваших существующих приложений календаря и контактов.

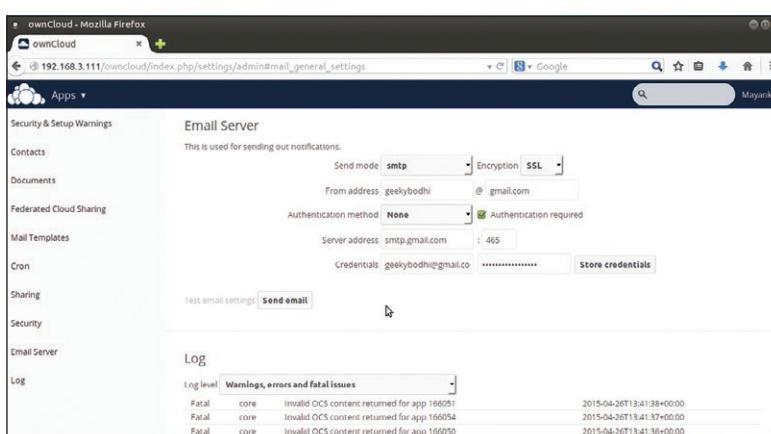
Теперь перейдите в *Contacts* в *OwnCloud* и нажмите на **Import Contacts** [Импорт контактов]. Во всплывающем окне нажмите на **Upload File** [Загрузить файл] и укажите ему на файл VCF. Когда контакты будут импортированы, можете синхронизировать их с вашим клиентом электронной почты с помощью ссылок CardDAV. Перейдите в раздел *Contacts* в *OwnCloud*, нажмите на иконку с шестеренкой внизу, наведите мышь на имя адресной книги, которую вы импортировали, и нажмите на иконку **Chain** [Цепочка]. Это выдаст CardDAV ссылку для данной адресной книги, которую вы можете передать на свой клиент рабочего стола или мобильной адресной книги.

## Синхронизируйте и делитесь календарем

Похожим образом *OwnCloud* управляет вашим календарем и задачами. Чтобы создать событие в вашем календаре, перейдите в приложение *Calendar*. Вы можете просмотреть календарь на весь месяц или на текущую неделю. Чтобы добавить новое событие, нажмите на соответствующую дату в календаре. Появится окно, предлагающее вам несколько опций для настройки события. Чтобы импортировать существующий клиент, просто загрузите файл iCAL на свой облачный сервер. Когда вы щелкнете по файлу в web-интерфейсе *OwnCloud*, сервер распознает файл и предложит импортировать его в существующий календарь или в новый. Выберите подходящую вам опцию.

Импортировав календарь, вы можете через *OwnCloud* делиться им с другими пользователями. Нажмите на иконку **Share Calendar** [Поделиться календарем] для соответствующего календаря. Раскроется меню, которое позволит вам выбрать пользователей или группу пользователей, с которыми вы хотите поделиться. Более того, как и с адресной книгой, *OwnCloud* может также синхронизировать ваши календари с настольным ПК и любыми мобильными приложениями, способными прочитать эту информацию со ссылок CalDAV. Чтобы получить ссылку CalDAV для вашего календаря, нажмите на кнопку с шестерenkами и затем на иконку **Chain** того календаря, который вы хотите синхронизировать. Отобразится ссылка, которую вы можете передать клиентам, чтобы они были синхронизированы с календарем *OwnCloud*. С помощью *OwnCloud* можно понаделать намного больше. Следуйте пошаговому руководству на стр. 95 (справа), чтобы расширить установку по умолчанию новыми приложениями, увеличив функциональность своего личного облака. [LXF](#)

» Прокрутите вниз до опций **Admin**, чтобы привязать *OwnCloud* к серверу электронной почты.



## Универсальный доступ

Реальное преимущество коммерческих облачных сервисов, таких как Dropbox, в том, что вы можете получить доступ к данным, хранящимся в них, с любого компьютера, подключенного к Интернету. Однако по умолчанию резидентная установка *OwnCloud* доступна только с компьютеров и устройств внутри локальной сети.

Но это не значит, что вы не можете получить доступ к вашему частному облаку из Интернета. Более хитроумное и дорогое решение — получить статический IP-адрес у своего интернет-провайдера и затем проделать дыры в брандмауэре своего

роутера или настроить Dynamic DNS в роутере или локальном компьютере. Впрочем, умнее будет использовать сервис туннелирования наподобие PageKite. Сервис использует модель «плати, сколько хочешь». Будучи некоммерческим пользователем, вы можете использовать сервис бесплатно, раз в месяц заполняя форму и сообщая PageKite, как вы используете сервис. Но он более чем стоит своих \$3 в месяц — минимальной платы, взимаемой с частных лиц.

Вначале надо установить PageKite. Запустите терминал и введите:

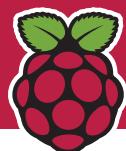
```
$ curl -s https://pagekite.net/pk/ | sudo bash
```

По завершении сделайте свой локальный web-сервер публичным, командой

```
$ pagekite.py 80 mycloudserver.pagekite.me
```

Не забудьте заменить mycloudserver на имя своего сервера *OwnCloud*. Теперь вы можете получить доступ к вашему личному экземпляру *OwnCloud*, перейдя на <http://mycloudserver.pagekite.me> с любого компьютера в любом точке мира. При первом запуске этой команды PageKite проведет вас по краткому процессу регистрации и спросит у вас ваш адрес электронной почты.

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



## Установим и активируем приложения

The screenshot shows the 'Apps - ownCloud' section of the OwnCloud web interface. It lists two applications: 'PDF Viewer 0.7' by Thomas Müller, Lukas Reschke (AGPL-licensed) and 'Provisioning API 0.3.0' by Tom Needham (AGPL-licensed). Both are marked as 'Official'. There are buttons for 'Show description...', 'Disable', and 'Enable only for specific groups'.

The screenshot shows the 'Documents - ownCloud' section. It displays a document titled 'Example.odt' with the content 'ownCloud Example Document'. The interface includes a toolbar with various document editing tools like bold, italic, underline, and font selection.

### 1 Подключенные приложения

Вы можете расширить вашу установку *OwnCloud* по умолчанию, добавив (или удалив) ряд приложений. Выберите выпадающее меню в верхней левой части интерфейса и нажмите на **Apps** [Приложения]. По умолчанию вам покажут список уже подключенного к вашей установке. Просмотрите этот список и прочтите описание, чтобы лучше понять их. Из этого раздела вы также можете отключить любое приложение.

The screenshot shows the 'Bookmarks - ownCloud' section. It lists several bookmarks with their addresses and creation dates. Tags are used to categorize them, such as 'Fedora', 'Linux', 'fresh', 'glorious', 'arch', 'fedora', 'linux', 'mint', etc. A 'Filterable Tags' sidebar allows for searching and filtering by tag.

### 3 Приложение Bookmarks

Другое продуктивное приложение, которое стоит включить — *Bookmarks* [Закладки]. Оно позволяет хранить и управлять закладками в вашем сервере *OwnCloud*. Закладки можно добавлять непосредственно или импортируя файл закладок из вашего web-браузера. У него также есть закладкограмма, которую вы можете добавить в закладки вашего браузера. Нажмите на закладкограмму, чтобы добавить сайт в список закладок *OwnCloud*.

The screenshot shows the 'Apps - ownCloud' section of the App Store. It lists several applications: 'Documents 0.9.2', 'OwnNotes 0.7.9', 'Notes 2.0.0', 'dashboard 1.6.2', and 'calendar 0.0.6'. Each item shows its name, developer, rating, and last update date.

### 5 Попробуйте еще

В добавок к приложениям, перечисленным в разделе **Apps** в вашем *OwnCloud*, есть и другие, которые вы можете установить с сайта. Прокрутите раздел **Apps** вниз и нажмите на ссылку **More Apps...** [Другие приложения]. Она отправит вас в магазин приложений *OwnCloud* на <http://apps.owncloud.com>. Вы можете скачать оттуда любое приложение и извлечь его в папку `/var/www/owncloud/apps` в Pi.

### 2 Приложение Documents

Это одно из самых полезных приложений. Оно может быть уже включено в вашу установку, а если нет, его можно получить из <https://apps.owncloud.com> (вместе со множеством других). Приложение *Documents* [Документы] позволяет вам редактировать и совместно работать над документами в разных форматах (включая DOC-файлы Word) внутри самого *OwnCloud*. За кулисами он использует *LibreOffice* из вашего Raspbian.

The screenshot shows the 'Calendar - ownCloud' section. It displays a monthly calendar for February 2016. The days of the week are labeled from Monday to Sunday. A specific event 'Chitzeni New Year' is shown for Saturday, February 13th.

### 4 Calendar и Contacts

Также к продуктивной категории относятся приложения *Calendar* и *Contacts*. После их включения даты и подробности могут добавляться интуитивно. Это позволяет загружать ваши существующие контакты и календари, которые можно синхронизировать с приложениями PIM из вашего *OwnCloud* (как объяснялось на уроке). Ряд приложений *OwnCloud* также имеют опцию включения их для определенных пользователей.

The screenshot shows the 'Storage Usage / Activity Charts - ownCloud' section. It displays two charts: 'Storage Usage' and 'Activity Charts'. The 'Storage Usage' chart shows disk usage statistics, while the 'Activity Charts' chart visualizes user activity over time.

### 6 Новости и занятость хранилища

Два других полезных приложения — *News* и *Storage Usage*. Первое — это программа для чтения RSS, а второе визуализирует объем хранения вашего облака в виде диаграммы. Когда вы скачаете их с сайта и извлечете в папку **apps**, вернитесь в раздел **Apps** в вашем *OwnCloud*. Эти и любые другие скачанные приложения перечислены в разделе **Not Enabled** [Не активны].

# Ответы

Есть вопрос по открытому ПО? Пишите нам по адресу [answers@linuxformat.ru](mailto:answers@linuxformat.ru), и мы найдем ответ.

» В этом месяце мы ответим на вопросы про...

- 1 Установку нашего ремикса Ubuntu
- 2 Начало работы с дистрибутивом Tails Linux
- 3 Проблемы обновления с Ubuntu LTS 14.04
- 4 Перенос Grub из MBR в загрузочный раздел
- 5 Подключение Android к Linux
- 6 Выбор между метками разделов и UUID

## 1 Установка ремикса Ubuntu

**В** Я понадеялся установить Ubuntu 15.10 с восемью рабочими столами, ради интереса. Однако сделать это удалось только с рабочим столом Unity, который я ненавижу. С диска LXF205-206 я загрузил Ubuntu 15.10 Desktop Remix, но во время установки получил следующее сообщение:

Installation Failed - The installer encountered an error copying files to the hard disk:

[Errno 2] No such file or directory: '/target/boot/vmlinuz-4.2.0-16-generic' This is often due to a faulty hard disk...

[Установка неудачна — инсталлятор обнаружил ошибку копирования на жесткий диск:

Нет такого файла... это бывает, когда жесткий диск неисправен

Я перезагрузился с диска LXF205-206 со стандартным рабочим столом Ubuntu 15.10, и установка прошла без ошибок. Как установить другой вариант?

Ixat, С форумов

**О** Нам больше не разрешают делать ремиксы для ISO-образов Ubuntu, если мы не удалим оттуда все ссылки на Ubuntu, а при этом не только теряется смысл упражнения, но и приходится компилировать большую часть программ с самого начала. Это вызвано изменениями в ограничениях торговой марки Ubuntu и означает, что нам нужно выбрать другой подход и создать внешнюю файловую систему, которая содержит дополнительное ПО и расположена поверх файловой системы ISO-образа Ubuntu. Эта возможность на самом деле была предназначена для того, чтобы Вы могли сохранять файлы и настройки на флешку и при этом пользоваться живой средой; но мы смогли применить ее и в наших целях. Однако установщик Ubuntu не подозревает об этой дополнительной файловой системе — и мы не рассчитывали, что он должен о ней знать — поэтому он терпит неудачу при установке с этого диска, и файлы ISO-образа, которые

он ожидал найти, во внешней файловой системе фактически были заменены.

Другими словами, ремикс для рабочего стола нужен только затем, чтобы попробовать различные рабочие столы в живом окружении. Чтобы воспользоваться ими со своего жесткого диска, следует установить стандартную систему Ubuntu и затем установить один из рабочих столов с помощью менеджера установки ПО. Это фактически ответ на жалобу о ремиксах, которые мы часто получаем: то, что они устанавливают всё, хотите Вы того или нет, хотя это не та причина, по которой мы выполнили изменение.

Мы рассказали об этой проблеме на домашней странице DVD, на странице Ubuntu DVD и на страницах DVD журнала. Мы часто включаем важную информацию в HTML-документацию диска или в страницы диска в журнале, или и туда, и туда — лучше ознакомиться с ней перед любой установкой.

## 2 Tails Linux

**В** У меня на ноутбуке Windows 7 есть несколько вопросов о Tails Linux. Обязательно ли после загрузки последней версии ISO-образа Tails Linux проверять его с помощью Gpg4win? Надо ли устанавливать Tails Linux с помощью Tails Installer? Можно ли, например, установить Tails Linux на мой внешний диск 1 ТБ с помощью Tails Installer?

В чем разница между I2P Browser и Tor Browser? Смогут ли они скрыть историю моих действий в Интернете от провайдера? Я бы хотел загрузить и установить последнюю версию Firefox Portable на флешку, на случай, если I2P Browser и Tor Browser не смогут открыть некоторые сайты. Нужно ли загружать его в I2P или Tor Browser? Tails Linux также включает копию Icedove Thunderbird, она действительно полностью идентична Mozilla Thunderbird?

Анонимный читатель

**О** Весьма приятно видеть, как новые пользователи Linux пробуют этот дистрибутив. Tails довольно сильно отличается от большинства дистрибутивов, и кривая его изучения будет иметь больший уклон, чем обычно, но это не означает, что не стоит его попробовать. Проверять образ не обязательно, зато так будет спокойнее. Устанавливать Tails тоже не обязательно, просто запишите ISO-образ на DVD или флешку. При использовании установщика на флешке создается защищенная область постоянного хранения данных, где Вы сможете сохранить настройки и другую информацию. Установщик очистит



» HTML-страницы на DVD и страницы DVD в журнале часто содержат важную информацию, с которой стоит ознакомиться перед установкой с этого DVD.

Вашу флешку и разобьет ее на разделы, поэтому убедитесь, что на ней нет ничего важного. Теоретически Вы можете использовать внешний диск, и он тоже будет очищен. Дистрибутив устанавливается на флешку или SD-карту. Установщик запускается из Tails, поэтому первый шаг — записать его на DVD и загрузиться с него, а если Вы используете наш LXFDVD — загрузитесь с него (мы уже проверили целостность образа).

И Tor, и I2P — анонимные сети, но реализованные по-разному. Tor считается более подходящим для «обычного» использования в Сети. Tor запускается по умолчанию, I2P запускается только в том случае, если Вы добавили add i2p в меню загрузки, нажав Tab при появлении меню, добавив i2p в появившийся список опций и нажав Enter для загрузки. И Tor, и I2P скрывают Вашу интернет-активность: провайдер видит только то, что Вы подключились к роутеру Tor (или I2P), но не окончательную точку назначения Вашего запроса. Причем она скрыта не только от провайдера, но и от промежуточных узлов Tor.

Если Вы хотите использовать Firefox Portable, лучше всего установить его на отдельную флешку, и неважно, через какой браузер или операционную систему это делать. Icedove построен на исходном коде Thunderbird, однако он был изменен разработчиками Debian, и это означает, что им больше нельзя назвать программу Thunderbird (аналогичным образом Firefox трансформировался в Icedove). Mozilla предприняла эти меры для защиты своих торговых марок.

## 3 Остаться ли с LTS?

**В** Я последовал вашему руководству еще в LXF185 и собрал себе компьютер с Linux с использованием по большей части

## Терминалы и суперпользователи

Мы часто предлагаем в качестве решения проблемы ввести те или иные команды в терминале. Хотя обычно то же самое можно сделать с помощью графических утилит дистрибутива, такие решения будут слишком конкретными (будут зависеть от дистрибутива). Команды в терминале более гибкие и — самое главное — ими можно пользоваться во всех дистрибутивах. Команды настройки системы часто нужно выполнять от имени суперпользователя, называемого также root. Существует два основных способа это делать, в зависимости от используемого дистрибутива. Во многих дистрибутивах, особенно в Ubuntu и его производных, перед командой можно написать sudo — при этом будет запрошено пароль пользователя, и ему будут предоставлены привилегии root только на время выполнения команды. В других дистрибутивах применяется команда su, для использования которой требуется ввести пароль root и которая предоставляет полный доступ root до того момента, пока вы не наберете logout. Если в вашем дистрибутиве используется su, запустите ее один раз и выполните любые заданные команды без предшествующей sudo.

именно тех компонентов, которые были представлены в обзоре [«Соберем Linux-ПК», стр. 30], за исключением процессора — я использовал Intel Core i7-4770. Я установил на него Linux Ubuntu 14.04 LTS с вашего диска. все загружается нормально, но при загрузке система отображается как Lubuntu; это нормально?

Я перешел на стадию, на которой должен обновиться до более свежей версии Ubuntu, но столкнулся с проблемой. Когда я в прошлый раз пытался обновляться на более старом компьютере, это превратилось в настоящую катастрофу, и я очень нервничал, поскольку не понимал, куда меня ведут команды. Менеджер обновления запускается с самого начала с четырьмя приложениями, не больше. Загрузка сразу же переходит на жесткий диск, и я не могу выбрать загрузку с оптического диска. Нужно ли переходить на 15.10 или лучше остановиться на 15.04 LTS?

Питер Рэтклифф [Peter Ratcliffe]

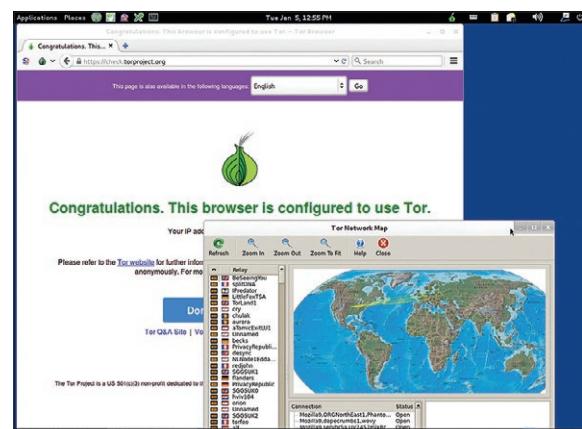
**О** Экран заставки Lubuntu — это неожиданный, но чисто косметический побочный эффект ремастеринга Ubuntu для ремикса рабочего стола. Существует два основных способа обновить существующую установку. Можно запустить обычный установщик Ubuntu, в котором есть возможность обновить существующую установку. Для этого он удаляет системные каталоги, но не каталог **home**, и затем выполняет свежую установку. Тогда Вы получаете новую версию без остатков предыдущей, но сохраняете все личные документы. Так как в число удаляемых каталогов входит **etc**, все системные настройки, а также дополнительные программы будут утеряны.

Альтернативный вариант — выполнить обновление из менеджера пакетов, который по сути запускает команду

```
$ sudo apt-get dist-upgrade.
```

Она обновит все пакеты до последней версии для нового дистрибутива и также обновит сами файлы дистрибутива до последней версии. Возможно, это не самый быстрый вариант, так как пакеты устанавливаются по отдельности, но он избавляет Вас от необходимости дополнительной настройки.

Обычно менеджер обновления постоянно изводит напоминаниями о появлении новой версии Ubuntu, но у Вас версия с долговременной поддержкой (LTS), поэтому он сообщит Вам только о доступности новой версии LTS, и этой версией будет 16.04 (релизы LTS выходят раз в два года). Поэтому в нем Вы видите лишь несколько приложений — те, которые были обновлены для версии 14.04. Если Вы обновите дистрибутив сейчас, то получите версию 15.10, которая не является релизом LTS и, возможно, Вам придется использовать первый способ. Что бы Вы ни выбрали, это будет



► Tails Linux — интересный, хотя и нестандартный способ познакомиться с Linux.

зависеть от того, что для Вас важнее — новейшее ПО или стабильность системы.

Можете воспользоваться 15.04 с DVD **LXF205**, но загрузить стандартный рабочий стол или воспользоваться вариантом установки из меню загрузки DVD. Как описано в вопросе «Установка ремикса Ubuntu» (см. стр. 96), установка из ремикса на жесткий диск не поддерживается. У большинства материнских плат есть меню загрузки, которое позволяет выбрать устройство для загрузки — из руководства по материнской плате Вы узнаете, какую клавишу удерживать при загрузке, чтобы его открыть — обычно это одна из функциональных клавиш или клавиша Esc.

### 4 Перенос Grub

**В** Мне надо установить Plop на жесткий диск ноутбука, но при этом я должен переместить Grub из главной загрузочной записи в раздел /boot дистрибутива, который я обычно запускаю (Linux Mint). Я нашел совет при запуске Grub нажать с, чтобы перейти в Bash и затем ввести root (hd0,4) для переноса Grub в раздел 5 на sda и ввести setup(hd0) для обновления. Вроде всё это просто, но когда я набираю эти команды,

»

## Коротко про...

# Буферы обмена

**М**ы все знаем, что такое буфер обмена: это область временного хранения информации, которая была скопирована или вырезана из одного приложения и должна быть вставлена в другое, или в то же самое, приложение. Однако далеко не все знают, что в настольном Linux есть несколько буферов обмена.

Тот, о котором мы уже рассказали — основной буфер обмена, он используется, когда вы что-то выбираете в приложении и нажимаете Ctrl+c. У дисплея X также есть свой буфер обмена, который часто называют буфером

выделения. Когда вы выбираете текст в X, он сразу же копируется в буфер выделения. После этого его можно вставить в текущее положение курсора мыши, нажав среднюю кнопку мыши (если у вашей мыши всего две кнопки, средняя кнопка эмулируется одновременным нажатием левой и правой кнопок).

Это очень просто и удобно, но может привести к путанице, поскольку это не один и тот же буфер обмена. Выберите текст в одном окне и нажмите Ctrl+v в другом — и вы получите не тот текст, который выбрали, а который последний раз копировался с помощью Ctrl+c/x: оба буфера

обмена полностью отделены друг от друга. Если в вашем рабочем столе есть менеджер буфера обмена, такой как Klipper в KDE или независимый CopyQ, вы сможете перебрасывать текст между двумя буферами обмена.

Также можно скопировать вывод команды оболочки прямо в буфер обмена с помощью команды xclip:

```
$ somecommand | xclip
```

```
$ somecommand | xclip -selection clipboard
```

Первая копирует вывод команды из буфера выделения X, вторая использует основной буфер обмена.

# Ответы

**Bash** отвечает can't find command [Команда не найдена]. Можете ли вы подтвердить, верны ли команды? В прошлом у меня уже были странные проблемы с *Bash* на ноутбуке.

**Кен [Ken]**

**O** При нажатии с открывается оболочка *Grub*, которая ничем не напоминает *Bash*. В ней можно выполнять только команды *Grub*. Команды, которые Вы вводили — это в самом деле команды *Grub*, но они относятся к старой версии *Grub*. Эти команды также устанавливают *Grub* в главную загрузочную запись диска. Linux Mint, как и почти все остальные дистрибутивы, использует *Grub 2*, который имеет другой набор команд.

Команды для *Grub 2* не только отличаются от команд для *Grub*, но их также можно запускать из обычного терминала *Bash*. Сначала нужно установить загрузчик в начало корневого раздела дистрибутива примерно такой командой:

```
$ sudo grub-install --force /dev/sda5
```

Параметр `--force` необходим при установке на раздел. Он заменяет две команды старой версии *Grub*, которые Вы нашли. Также нужно убедиться, что для раздела установлен флаг `bootable` (загрузочный), чтобы BIOS знал, где искать загрузчик. Этот флаг можно установить с помощью любой утилиты, которую Вы предпочитаете, например, *GParted* или *fdisk*. Однако система все равно сначала будет искать загрузчик в главной загрузочной записи, поэтому по-прежнему загрузится более ранняя установка *Grub*. В зависимости от BIOS Вы сможете выбрать в меню загрузки конкретный раздел, чтобы посмотреть, как он работает. В противном случае Вы сможете удалить загрузчик из главной загрузочной записи командой `dd`, но будьте осторожны, поскольку там также хранится таблица разделов. Поэтому сначала нужно создать резервную копию главной загрузочной записи командой:

```
$ sudo dd if=/dev/sda of=oldmbr.img bs=512 count=1
```

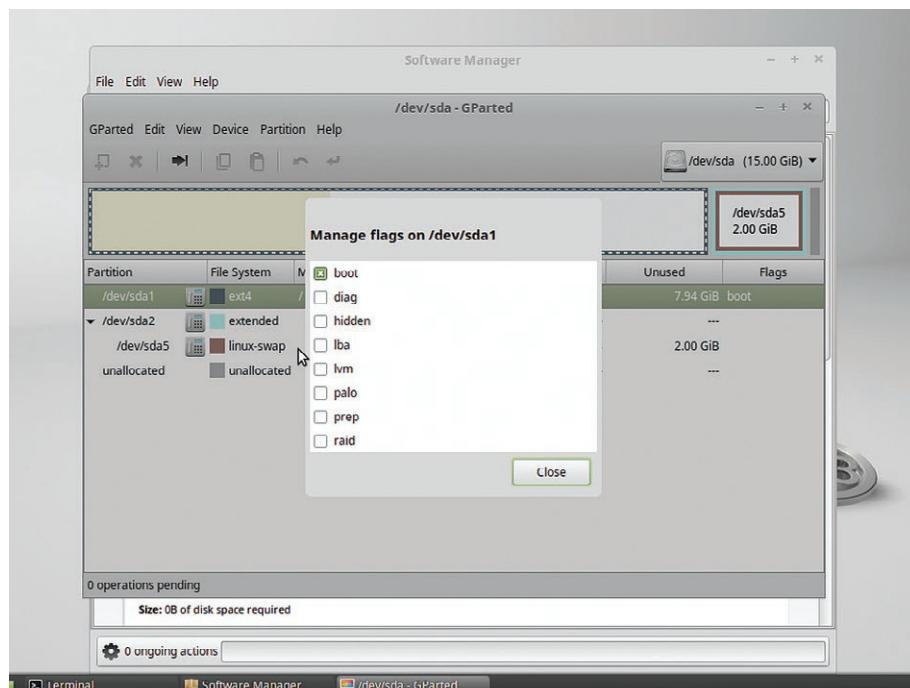
Если что-то пойдет не так, Вы сможете восстановить главную загрузочную запись, поменяв в этой команде местами содержимое параметров `of` и `if`. Затем удалите *Grub* командой

```
$ sudo dd if=/dev/zero of=/dev/sda bs=446 count=1
```

Обратите внимание на разные размеры. Первая команда выполняет резервное копирование всей MBR, на всякий случай, а вторая обнуляет только часть, содержащую загрузчик. Также можно установить в MBR *Plop* или любой другой загрузчик, чтобы перезаписать *Grub*. Загрузка *Grub 2* из раздела не поддерживается, отсюда и необходимость использовать `--force`, так что это не самый надежный подход. Возможно, лучше установить *Plop* и дайте ей обо всем позаботиться.

## 5 Android в Linux

**B** На моем настольном компьютере установлены Ubuntu 15.10 и Mint 17.2. Они прекрасно работают — за исключением того случая, когда я подключаю свой планшет Medion через USB, и он просто не распознается ни одной из этих систем. При этом, когда я подключаю флешку с фотографиями, обе системы



Если вы перемещаете *Grub* из главной загрузочной записи вашего диска на раздел, надо установить загрузочный флаг для этого раздела.

распознают ее. Я позвонил производителям, и они говорят, что подключение плашета будет работать только в Windows. Так как у меня нет Windows и я ничуть не собирался ее устанавливать, можете сказать, как обойти эту проблему?

**Джон Джексон [John Jackson]**

**O** Устройства Android монтируются не так, как флешки. Причина состоит в том, что только одна ОС может монтировать файловую систему в один момент времени, в противном случае изменения, внесенные одной ОС, могут разрушить другую. В более ранних устройствах Android Вас заставляли размонтировать локальное хранилище перед его монтированием другим компьютером, что неудобно с внешним хранилищем и невозможно с внутренним хранилищем, используемым ОС. Теперь в Android используется MTP, это своего рода сетевая файловая система для подключений USB, предназначенная специально для монтируемых устройств, таких как смартфоны и медиа-проигрыватели.

Вам нужно установить файловую систему (я добился определенных успехов с `jmt�fs` — <https://github.com/JasonFerrara/jmt�fs>), затем подключить свое устройство и смонтировать его из терминала командой

```
$ jmt�fs MOUNTPOINT
```

где `MOUNTPOINT` — пустой каталог. При запуске от имени обычного пользователя Вы сможете просматривать содержимое своего планшета с компьютера, копировать и удалять файлы. Закончив, размонтируйте таблицу командой

```
$ fusermount -u MOUNTPOINT
```

`Jmt�fs` и некоторые альтернативные файловые системы используют уровень файловой системы FUSE ядра Linux. Возможно, Вы захотите попробовать другие, чтобы разобраться, что лучше

подойдет для Вас, для этого наберите `"mt�fs"` в строке поиска менеджера пакетов.

У разных пользователей успех с различными реализациями MTP был переменным, отсюда и предложение попробовать альтернативы. Одна из альтернатив — вообще не использовать MTP, но использовать вместо него SSH. Этот вариант также удобнее, поскольку использует беспроводное подключение. Установите на свой планшет *SSH Helper* и запустите программу, убедившись, что планшет и компьютер подключены к одной сети. В окне настройки *SSH Helper* будут показаны IP-адрес и номер порта, на котором запущен SSH (номер порта по умолчанию — 2222). Задайте в этом окне подходящий пароль. Затем установите *SSHFS* на свой компьютер, и теперь Вы сможете смонтировать любой компьютер, к которому можно подключиться по SSH, например, такой командой:

```
$ sshfs -p 2222 IP-ADDRESS:/storage/emulated/0 MOUNTPOINT
```

Возможно, Вам придется изменить путь к внутреннему устройству хранения: приведенный выше соответствует последним устройствам Nexus. Также убедитесь, что точка монтирования, которую Вы используете, существует и пуста. Теперь ваше устройство смонтировано на компьютере, как и в старые добрые времена USB-флешек. Когда закончите, размонтируйте его командой

```
$ fusermount -u MOUNTPOINT .
```

## 6 Метки или UUID?

**B** У меня был довольно странный опыт с последней установкой Ubuntu 15.10 на новом твердотельном диске. Я создал отдельные разделы для системы и для `home`. Я выбрал использование существующего раздела подкачки на обычном диске. Для установки я воспользовался флешкой, созданной с помощью *UNetbootin*.

Всё было хорошо, пока я не перезагрузился, что заняло целую вечность. Оказалось, что установщик указал раздел подкачки в `fstab` в формате `/dev/sdX`, тогда как два других раздела были указаны в формате `UUID`. Так как флешки больше не было, раздел `/dev/sdX` изменился, и поэтому раздел подкачки не был найден. Я знаю, что для указания дисков в `fstab` рекомендуются `UUID`, но решил сделать по-другому: `LABEL=####`. Через несколько минут мой раздел подкачки получил метку `SWAP`, в `fstab` он стал указан как `LABEL=SWAP`, и перезагрузка была выполнена моментально (`SSD` — прекрасная штука).

**Есть ли недостатки у подхода `LABEL=SWAP`? Мне он кажется намного проще, особенно потому, что я хочу переразбить жесткий диск и просто хочу убедиться, что где-то в системе нет раздела подкачки с меткой `SWAP`. При использовании `UUID` мне пришлось бы найти новый `UUID` и снова изменить `fstab`.**

Джон Парр [John Parr]

Подозреваю, что Ваша проблема началась с использования UNetbootin, так как тогда Вы загружались с флешки USB, которая забрала узел диска `/dev/sda`. Сейчас в большинстве дистрибутивов Linux нет нужды

обращаться к `UNetbootin`, так как они используют гибридные ISO-образы. Их можно просто скопировать на флешку командой `dd`, например:

```
$ dd if=ubuntu-15.10-desktop-amd64.iso of=/dev/sdb bs=1M
```

При загрузке с этой флешки она распознается как привод USB CD/DVD, и ей назначается имя узла `/dev/sr0`, при этом `sda` освобождается для Вашего жесткого диска. Однако Вы упомянули проблему с использованием узлов устройств в `fstab`, вот поэтому дистрибутивы стараются использовать `UUID`, чтобы избежать путаницы.

Метки — хорошая альтернатива, они лучше читаются, но могут вызвать проблемы, если Вы перейдете на другой компьютер, на котором уже есть такие же метки. Чтобы этого избежать, я обычно добавляю к метке имя хоста вместо использования простых `swap`, `home` или `root`.

Заглянув в `/dev/disk`, Вы найдете несколько каталогов, например, `by-label` и `by-uuid`, которые содержат символические ссылки с меткой и `UUID` на пути к реальным устройствам. Вы также можете найти записи для `partlabel` и `partuuid`. Любую из них можно использовать в `/etc/fstab`, и все они будут одинаково эффективны. Единственный реальный выбор между ними касается схемы именования, с которой Вам наиболее комфортно. [LXF](#)

## Помогите нам помочь вам

Ежемесячно к нам поступает несколько писем, на которые мы не в состоянии ответить, поскольку проблема описана в них с недостаточной полнотой. Чтобы дать вам наилучший ответ, нам необходимо получить как можно больше информации о проблеме.

Если у вас появляется сообщение об ошибке, приведите его текст в точности и опишите конкретные условия, когда оно появляется. При возникновении проблемы с устройствами перечислите нам все устройства, которые у вас установлены.

Если Linux в вашей системе запущен, вы сможете применить для этого превосходную программу `Hardinfo` (<http://sourceforge.net/hardinfo.berlios>) — она сохранит подробную информацию об устройствах и о состоянии системы в HTML-файле, который вы сможете приложить к своему письму, отправляемому нам.

Не уступающий в удобстве альтернативный вариант — `lshw` (<http://ezix.org/project/wiki/HardwareLister>). Одна из указанных программ непременно должна быть включена в ваш дистрибутив (а иногда и обе).

Если у вас нет желания или возможности их установить, выполните в терминале от имени root приведенные ниже команды и приложите сгенерированный ими файл `system.txt` к письму. Это окажет неоценимую помощь в диагностике вашей проблемы.

```
uname -a > system.txt
lspci >> system.txt
lspci -vv >> system.txt
```



## Коротко про...

# Поиск ресурсоедов

Что вы делаете, если вам кажется, что компьютер работает медленнее, чем должен? Мы имеем в виду не скорость загрузки Интернета, а общее быстродействие рабочего стола. Возможно, есть какой-то процесс, потребляющий слишком много циклов процессора, памяти или и то, и другое. Это можно проверить с помощью различных утилит. Среди них есть и несколько графических, но стандартная утилита — `top`. Она запускается в терминале, но использовать ее можно везде — в том числе и через SSH — и она является частью инструментария, который вы найдете в любом дистрибутиве Linux.

Запустите `top` без аргументов, и вы увидите список процессов, упорядоченный по потреблению процессора, с некоторой информацией о системе в верхней части окна. После общей информации в третьей строке отображаются данные о потреблении процессора, разбитые на несколько колонок: `us` (пользовательские процессы), `sy`

(системные процессы), `ni` (процессы, запущенные со значением nice), `id` (простаивающие процессы) и `wa`. В последней колонке отображается время, потраченное на ожидание, обычно ожидание ввода/вывода — вы увидите, как оно возрастает при запуске процесса, интенсивно работающего с диском. Даже несмотря на то, что у вас скорее всего будет многоядерный процессор, по умолчанию `top` усредняет использование ядер к одному набору показателей. Чтобы увидеть использование для каждого из ядер, нажмите клавишу `1`.

В следующих строках отображается информация об использовании памяти и раздела подкачки, за ней следует список процессов. По умолчанию он сортируется по использованию процессора, и наиболее интенсивные программы отображаются в верхней части списка. В нем отображается сколько процессов, сколько умещается в вашем окне терминала. Если вы запустите `top`, чтобы узнат, почему система работает так

медленно, вы можете увидеть, что процесс в верхней части списка использует очень много циклов процессора. Этот процесс легко завершить, нажав `k`. После этого `top` запросит PID (идентификатор) процесса, который нужно завершить. Они отображаются в левом столбце, но `top` по умолчанию любезно сама подставляет идентификатор верхнего процесса в списке. Нажмите `Enter`, и команда спросит, какой сигнал нужно отправить. По умолчанию это `15`, или `SIGTERM`, который сообщает процессу о том, что нужно корректно завершиться. Если сделать этого не удается, что может иметь место с заблокированным процессом, попробуйте заменить сигнал на `9` (`SIGKILL`) для немедленного завершения процесса.

`Top` по умолчанию сортирует процессы по использованию процессора, но отображаемые столбцы и порядок сортировки можно изменить. Нажмите `f` для входа в окно управления полями. В нем можно навести курсор на поле и нажать пробел, чтобы отобразить и скрыть столбец, или нажать `s`, чтобы отсортировать по этому

任务	状态	CPU 使用	内存 使用	磁盘 I/O	网络 I/O
9428 sambred	20	0	36920	25532	3690 99.7 0.7 12700 20
9889 root	20	0	21812	2392	2446 R 9.7 0.1 12700 20
15020 root	20	0	10000	10000	10000 0.0 0.0 0.0 0.0 0.0
17751 transmission	20	0	351772	50256	3985 S 0.3 1.3 378 12 20
23779 root	20	0	72240	10000	10000 0.0 0.0 0.0 0.0 0.0
1 root	20	0	4204	1224	1239 S 0.0 0.0 0.0 0.0 0.0
2 root	20	0	0	0	0 S 0.0 0.0 0.0 0.0 0.0
3 root	20	0	0	0	0 S 0.0 0.0 0.0 0.0 0.0
4 root	20	0	0	0	0 S 0.0 0.0 0.0 0.0 0.0
5 root	0	-20	0	0	0 S 0.0 0.0 0.0 0.0 0.0
7 root	0	-20	0	0	0 S 0.0 0.0 0.0 0.0 0.0
8 root	0	-20	0	0	0 S 0.0 0.0 0.0 0.0 0.0
9 root	0	-20	0	0	0 S 0.0 0.0 0.0 0.0 0.0
10 root	0	-20	0	0	0 S 0.0 0.0 0.0 0.0 0.0
11 root	20	0	0	0	0 S 0.0 0.0 0.0 0.0 0.0
12 root	0	-20	0	0	0 S 0.0 0.0 0.0 0.0 0.0
13 root	0	-20	0	0	0 S 0.0 0.0 0.0 0.0 0.0
14 root	0	-20	0	0	0 S 0.0 0.0 0.0 0.0 0.0
15 root	20	0	0	0	0 S 0.0 0.0 0.0 0.0 0.0
16 root	0	-20	0	0	0 S 0.0 0.0 0.0 0.0 0.0

► Нажмите `1`, чтобы увидеть каждое ядро по отдельности.

полю. Нажмите `Esc`, чтобы вернуться обратно к списку процессов. Также можно изменить порядок отображаемых здесь столбцов. Есть несколько ярлыков для сортировки по распространенным полям: `M` — память, `P` — процессор и `T` — время работы.

Также при запуске `top` можно указать поле сортировки в командной строке, указав параметр `-o`, за которым следует название поля, например:

```
$ top -o %MEM
```

Настройки `top` по умолчанию находятся в файле `~/.toprc`. Этот файл не нужно редактировать напрямую — надо лишь нажать `W` при запущенной команде `top`, и текущая схема будет сохранена.



Лучшие в мире программы  
с открытым кодом

# LXF HotPicks

Sprut.io » Mumble » Let's Encrypt » Qemu » Mixxx » MP3fs » Boxes  
» SuperTux » BiniAx » Zeal » PeaZip



Александр Толстой

сломя голову носится по островкам  
Интернета, как большой ребенок в магазине сластей, и выхватывает для вас  
лучшие программы с открытым кодом.

## Менеджер файлов

# Sprut.io

Версия: git Сайт: <http://bit.ly/Sprutio>

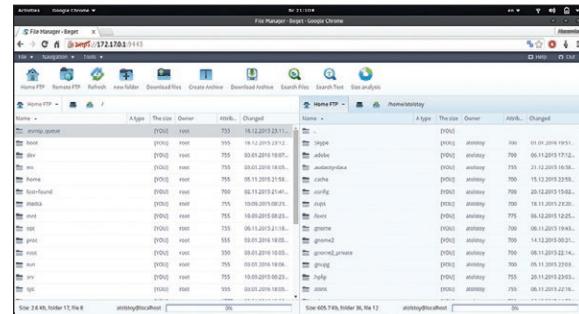
**В**озможность просматривать свою файловую систему — краеугольный камень работы с персональным компьютером, и она существует уже долгое время, начиная с древних консолей UNIX. Традиционные менеджеры файлов не являются передовой технологией, пусть даже в наше время они и выглядят модно — и «материално»; так почему бы не попробовать обновиться до технологии следующего поколения и не применить для управления своими файлами клиентскую архитектуру?

*Sprut.io* — выдающееся приложение, реализующее вышеописанный подход и очень удобное для управления файлами

на удаленных серверах, передачи данных между удаленными компьютерами и работы с размещенными сайтами, и т.д. *Sprut.io* некоторое время был проприетарной программой, но недавно его разработчик, компания Beget Inc., открыла код и разместила его в постоянном репозитории Github.

*Sprut.io* был написан с помощью Python и среди Tornado, но выглядит, как классический менеджер файлов с двумя панелями,

**Удобно для управления файлами на удаленных серверах.**



Используйте опцию страницы Translate this [Перевести] в *Sprut.io*, чтобы насладиться английским интерфейсом...

поддержкой перетаскивания и вкладками. Кроме того, в нем есть: встроенный текстовый редактор с выделением синтаксиса; транскодер символов; просмотрщик изображений; клиент FTP; менеджер архивов; и некоторые другие полезные аксессуары. Даже если вам не приходится заниматься поддержкой крупной сети, а всего лишь работать с домашним роутером и несколькими клиентами, *Sprut.io* все равно очень полезен и привносит новый уровень удобства в LAN. Он намного дружелюбнее к пользователю, чем соединение через SSH с MC (при всем уважении к этому методу).

Установка *Sprut.io* может оказаться нетривиальной, но, к счастью, существует готовый к использованию образ Docker. Возможно, вы представления не имеете, как устанавливать Docker, если раньше вам не приходилось этого делать, однако тут всё довольно просто:

```
$ wget -qO- https://get.docker.com/ | sh
```

По завершении соединитесь с сервером через SSH, перейдите в директорию, где вы хотите установить *Sprut.io*, и введите следующее:

```
$ wget https://raw.githubusercontent.com/
```

```
LTD-Beget/sprutio/master/run.sh
```

```
$ chmod +x run.sh
```

```
$ ./run.sh
```

Установка может потребовать до 10 минут, в зависимости от настроек вашего оборудования и сети. И, наконец, вы сможете запустить *Sprut.io*, посредством перехода на [https://your\\_server\\_IP:9443](https://your_server_IP:9443) в своем браузере. Наслаждайтесь!

## Исследуем интерфейс *Sprut.io*

### Верхняя панель инструментов

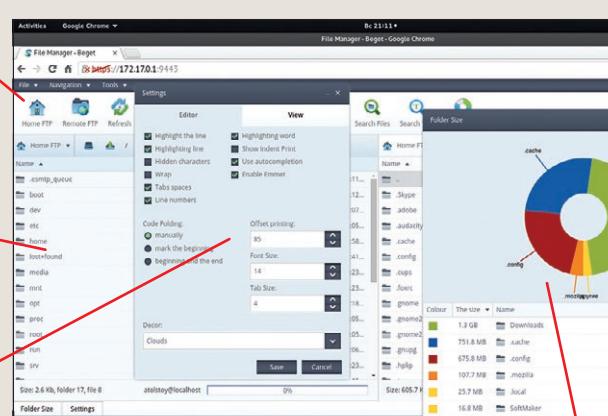
На верхней панели размещены значки большего размера, для часто выполняемых действий и мест.

### Файлы и директории

Просматривайте локальные и удаленные файловые системы точно так же, как с привычным менеджером файлов.

### Настройки

Изменяйте внешний вид и работу режима View и встроенного редактора.



### Ярлыки внизу

Часто выполняемые действия имеют привязки к клавишам, что может сэкономить ваше время.

### График использования диска

*Sprut.io* имеет собственный клон *Baobab* или *Filelight* для визуализации потребления диска, в виде секторной круговой диаграммы.

Клиент VoIP

# Mumble

Версия: 1.2.11 Сайт: [www.mumble.info](http://www.mumble.info)

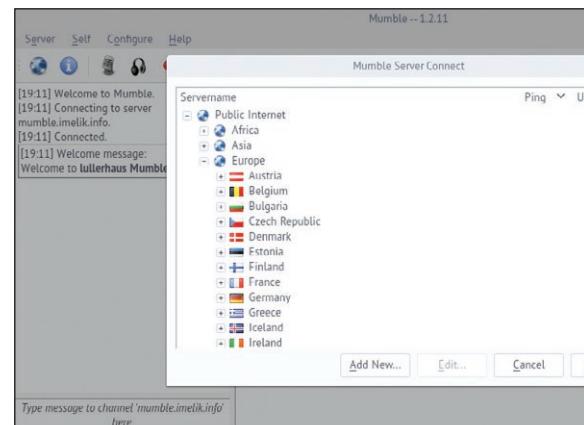
**K**ак ни удивительно, общение между игроками в современных 3D-стрелялках и командных играх до сих пор остается нестабильной и вто-ростепенной функцией. Большинство решений этой проблемы — корявые обходные пути, и многие геймеры используют платные серверы *TeamSpeak* или возятся со *Skype*, и хотя эти решения хорошо спра-вляются со своей работой, они не уважают вашу свободу.

*Mumble* — это свободная (во всех смыс-лах) программа общения с другим челове-ком. Вы можете применять ее как телефон VoIP и для обмена текстовыми сообщениями, но в первую очередь *Mumble* предназна-чена для игр.

*Mumble* уже настроена для высокоско-ростного сетевого соединения с малым значением задержки и поддерживает мно-жество круtyх функций — например, объ-емный звук. Это означает, что *Mumble* на-страивает баланс между правой и левой колонкой или наушниками в соответсвии

с положением игроков, которые разгово-ривают с вами. *Mumble* получает эту ин-формацию прямо из игры (посылая кон-текст камера/аватар), и, конечно, чтобы объемный звук работал, оба игрока долж-ны использовать *Mumble* или совмести-мые приложения.

При первом запуске *Mumble* удобный мастер любезно проведет вас по начальной настройке, которая определит ваше зву-ковое оборудование и позволит протести-ровать воспроизведение и запись звука. По завершении вы перенесетесь в главное окно *Mumble*, очень похожее на обычный чат IRC. Слева находится область для соо-бщений, а справа — список пользователей. Первое, что вам надо сделать — нажать на значок «глобус» на панели инструментов



Есть множество общественных серверов, не требующих аутентификации.

и выбрать подходящий сервер в вашей стране (вы должны быть уже зарегис-трированы на нем). Войдя и убедившись, что и текст, и звук настроены корректно, мож-ете запускать игру и начинать хамить чле-нам своей команды.

**Mumble настроена для сетевых соединений с малой задержкой.**

Источник сертификатов

# Let's Encrypt

Версия: 0.1.1 Сайт: <https://letsencrypt.org>

**L**et's Encrypt — это инструмент для организации безопасного досту-па к сетевым ресурсам (доменам), пред назначененный для хостов, системных администраторов и в общем для всех, кто управляет web-сервером, неважно — уров-ня предприятия или домашней страницей в локальной LAN. Выпускает Let's Encrypt Internet Security Research Group (ISRG), тех-нологическая фирма из Калифорнии, ко-торая финансируется и некоммерческими организацами (например, Mozilla), и про-мышленными гигантами (например, Cisco).

Хотя в прошлом существовали и дру-гие провайдеры свободных сертификатов SSL, у них были определенные ограничения и оговорки: например, невнятная мо-дель лицензирования для коммерческого использования или значительные затраты на техобслуживание. Let's Encrypt является лучшим решением для надежного и прозрачного безопасного доступа к web-доме-ну. Протокол, на котором работает сервис,

называется ACME, и он тоже имеет откры-тый код и разработан рабочей группой In-ternet Engineering Task Force (IETF).

Let's Encrypt — это классический инст-румент клиент—сервер, то есть компьютер, на котором вы удостоверяете свой домен, являетсся клиентом, а сервис ACME явля-ется сервером авторизации. В качестве ил-люстрации, запустим клиент Let's Encrypt, чтобы вручную проверить взятый для при-мера сервер:

```
$ letsencrypt --text --email recovery@example.com --domains www.example.com,example.com,foo.example.com --agree-to-s --renew-by-default --manual certonly
```

В этой команде мы предоставили некий фиктивный адрес электронной почты для



На сайте проек-та есть подробное объяснение, как ра-ботает Let's Encrypt.

восстановления потерянной информации по учетной записи и перечислили псевдо-имена доменов, которые клиент Let's Encrypt может авторизовать. Эта секретная информация будет храниться в директо-рии, подобной /var/www/html/.well-known/acme-challenge, так что проверьте, сущес-тует ли она (или создайте ее командой mkdir), и, наконец, переместите туда ин-формацию, предоставленную командой letsencrypt. Когда вы пристроите эти дан-ные к месту, сервер ACME проверит сайт. Если конфиденциальная информация сов-падает, Let's Encrypt предоставит серти-фикат. Когда проверка будет выполнена, в /etc/letsencrypt появится настройка для домена вместе с ключами, сертификатом и использованным запросом на подпись.

**Лучшее решение для безопасного доступа к web-домену.**

## Программа виртуализации

# Qemu

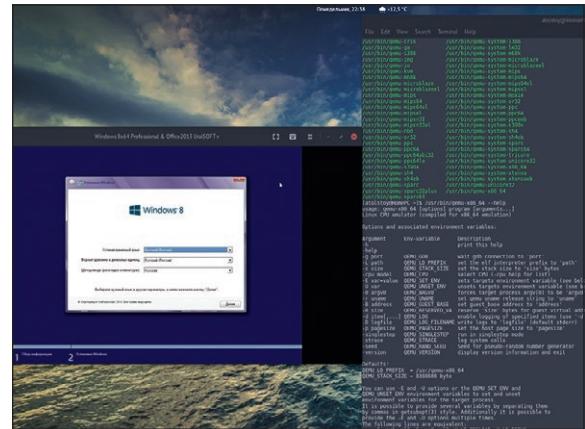
Версия: 2.5 Сайт: <http://wiki.qemu.org/Download>

**Q**emu — это т.н. «архитектурный эмулятор», который использует гипервизор KVM и может создавать виртуальные ПК на очень разных архитектурах. Для обычных пользователей настольных ПК *Qemu* является хорошей альтернативой *VirtualBox* (поддерживающей Oracle) или *VMWare Player* (с закрытым кодом), и обеспечивает высококачественную технологию виртуализации с компонентами, имеющими открытый код. В последние месяцы *Qemu* быстро развивается. В ноябре 2015 г. версия 2.4 добавила, среди прочего, драйвер VirtIO-GPU DRM для 2D-графики, который разогнал производительность гостевых машин с *Qemu*. В *Qemu* 2.5 поддержка графики достигла еще одного долгожданного рубежного момента — значительного улучшения модуля VirtIO-gpu, который позволяет гостевым системам использовать ускорение OpenGL, предоставленное системой хоста. Это решило проблему узких мест производительности системы, которая долгие годы

влияла на гостевые системы *Qemu* — может, на эмуляцию серверов и не столь значительно, но уж факт, что явно замедляя гостевые компьютеры.

Есть и множество других заметных изменений — например, live-миграция, использующая метод Postcopy (вы можете передавать гостевые системы на другой хост, не останавливая их), лучшая производительность драйверов и новые возможности горячего подключения, и т.д. Смело можно сказать, что если вы запускали виртуальные ПК с *Qemu* в прошлом, то после обновления до версии 2.5 заметите улучшение производительности и более широкий выбор функций.

Технически *Qemu* является утилитой командной строки, и во многих системах



Создать виртуальный ПК с помощью *Qemu* легко и в командной строке, и с помощью Boxes.

вы найдете массу исполняемых файлов, соответствующих маске `/usr/bin/qemu-*`. Каждый файл относится к определенной архитектуре, которую он эмулирует, так что вам, вероятно, придется использовать `/usr/bin/qemu-arm` для эмуляции каких-нибудь наладонников на базе ARM или `/usr/bin/qemu-x86_64` для запуска современного ПК на основе Intel. Однако есть более дружелюбный к пользователю способ познакомиться с *Qemu*, и это — *Gnome Boxes*, отличный графический интерфейс для Libvirt, который, в свою очередь, является драйвером *Qemu*. *Gnome Boxes* имеется в десятках дистрибутивов Linux, и неважно, какой дистрибутив вы выберете.

Позволяет гостевым ОС использовать ускорение OpenGL хоста.

## Аудиомикшер и редактор

# Mixxx

Версия: 2.0 Сайт: <http://mixxx.org>

**Э**та программа — для творческих личностей, тех, кому интересно превращать популярные мелодии в танцевальные треки, или которые имеют контроллер Denon или American Audio DJ и ищут, что бы такого с ними сделать. Не ищите больше, у вас есть *Mixxx*. Это невероятно умное приложение с открытым кодом, хорошо продуманное и доступное даже для новичков. Впервые увидев окно *Mixxx*, вы, возможно, струхнете, что заблудитесь в десятках крошечных кнопок управления, но уже через несколько минут сообразите, что все они размещены в весьма разумном порядке.

У *Mixxx* три деки, и на каждой из них есть размечаемая проматываемая волновая форма, которая отмечает такты и ключевые точки, вместе с волновой формой всей песни, для быстрого поиска. Имеется также очень удобный автоматический инструмент определения BPM (ударов в минуту), который анализирует и считывает

темп вашей песни, и очень мощный движок микшера, способный даже микшировать треки без вмешательства пользователя (как Auto DJ).

В нижней части интерфейса *Mixxx* — широкий плей-лист, неотличимый от области плей-листа обычного аудиоплейера. Слева находится дерево категорий, и хотя оно довольно маленько, вы сможете часто его использовать для переключения по вашей музыкальной библиотеке, записям или плей-листам.

Остальная часть окна *Mixxx* тоже отнюдь не страшная, она просто так выглядит из-за мрачной темы. Окно структурировано в четыре цепи микшера, дисплей волновой формы с управлением громкостью



*Mixxx* — удивительное приложение профессионального уровня с простой настройкой и полностью открытым кодом.

ниже, управлением скоростью и подробной информацией о треке. Обратите внимание на меленъкие кнопки управления над основной волновой формой — они позволяют быстро индивидуально настроить управление, включать/выключать эквалайзер и отображать относящиеся к винилу кнопки. Кроме того, *Mixxx* поддерживает массу звуковых периферийных устройств — и для ввода, и для вывода. Перейдите в Избранное программы (Ctrl+p) и нажмите на кнопку Query Devices [Устройства по запросу], чтобы посмотреть, что присоединено к вашему ПК. Вы можете проделывать всякие крутие штуки со своим цифровым виниловым деком, даже и не прикасаясь к нему!

Поддерживает массу звуковых устройств для ввода и вывода.

## Конвертор аудио

# MP3fs

**Версия:** 0.91 **Сайт:** <http://khenriks.github.io/mp3fs>

**M**ногие аудиофайлы и ценители высококачественного звука Hi-Fi предпочитают неожиданные форматы файлов или, по крайней мере, форматы без потерь, а не MP3, и основным выбором большинства, вероятно, является FLAC (Free Lossless Audio Codec).

Музыкальные треки, сжатые с помощью FLAC, довольно небольшие, но при этом в них нет ни в чем особых потерь. Иногда возникает необходимость перекодировать файлы FLAC в MP3, просто потому, что ваше целевое устройство или приложение не поддерживает прелестей FLAC. Но как, по-вашему, выглядит транскодер форматов файлов? Как инструмент командной строки? GUI со множеством вкладок и кнопок? Меню сервиса для вашего менеджера файлов? Или, может быть, тяжеловесный редактор волновых форм?

Мыслите иначе, потому что вот сейчас мы будем конвертировать звуковые файлы FLAC в MP3 с помощью файловой системы FUSE.

MP3fs предлагает оригинальный подход к конвертированию файлов. Вы монтируете свою исходную директорию с файлами FLAC как файловую систему MP3fs, затем переходите в целевую директорию, смонтированную как файловая система с типом 'mp3fs', например:

```
$ mp3fs -b 192 /mnt/music /mnt/mp3 -o allow_other,ro
```

где `/mnt/music` хранит ваши исходные файлы FLAC, 192 — битрейт, а `/mnt/mp3` — целевая директория. Когда вы будете использовать FUSE впервые, раскомментируйте переменную `user_allow_other` в `/etc/fuse.conf`, это чтобы разрешить пользователям, не являющимся root, монтировать и размонтировать файловые системы FUSE. После выполнения команды `mount` вы найдете

## Конвертируйте звуковые файлы FLAC в MP3 с помощью FC FUSE.

### MP3FS(1) Manual Page

#### NAME

`mp3fs` - mounts and transcodes FLACs to MP3s on the fly

#### SYNOPSIS

```
mp3fs [OPTION]... FLACDIR MP3DIR
```

#### DESCRIPTION

The `mp3fs(1)` command will mount the directory `FLACDIR` on `MP3DIR`. Thereafter, accessing `MP3DIR` will show the contents of `FLACDIR`, with all FLAC files transparently renamed and transcoded to MP3 format upon access.

#### OPTIONS

```
-b, --bitrate=RATE
```

Set the bitrate to use for encoding. Acceptable values for RATE are any which are allowed by the MP3 format. According to the manual for LAME, this means:

For sampling frequencies of 32, 44.1, and 48 kHz, RATE can be among 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, and 320.

For sampling frequencies of 16, 22.05, and 24 kHz, RATE can be among 8, 16, 24, 32, 40, 48, 56,

➤ Откройте опции настройки монтирования для MP3fs на очень полезной странице справочника.

целевую директорию с размещенными в ней файлами, такими же, как и в исходной директории, но с расширениями `.mp3`. Когда вы попытаетесь воспроизвести файлы, ваша система будет обращаться с ними, как с обычными MP3-файлами, хотя вы, возможно, заметите некоторую задержку перед воспроизведением — MP3fs нужно время и дополнительные лошадиные силы для перекодирования файлов на лету с помощью LAME. Если вам нравится идея, добавьте соответствующую строку в свою `/etc/fstab` с помощью

```
mp3fs#/mnt/music /mnt/mp3 fuse allow_other,ro,bitrate=192 0 0
```

## Инструмент командной строки

# Boxes

**Версия:** 1.1.2 **Сайт:** <http://boxes.thomasjensen.com>

**П**ри каждой встрече с изображениями ASCII трудно удержаться от мысли: «Полная дурь, но я это все равно использую, поскольку оно до смешного мило и очень греет душу!» Прошлый раз был, наверное, с Ubuntu Hollywood Technodrama [HotPicks, стр. 105 LXF195], где был явный передоз ASCII-картинок, мигающих во множестве терминалов.

На сей раз ASCII используется для декодирования пословиц, цитат и текста в целом в крошечной утилите под названием `Boxes` (не путайте с `Gnome Boxes`). Этот аккуратный инструментик размещает вокруг текста блоки псевдографики, позволяя пользователю выбрать дизайн блока из 51 варианта, предлагаемого в `Boxes` по умолчанию.

Хотя это и вполне очевидно, но приятная вещь в дизайне блоков — каждый из них создается вручную из обычного текста, позволяя необычайно легко создавать свой собственный дизайн. Вы можете начать с просмотра `/usr/share/boxes/boxes`,

чтобы увидеть, что вам там предлагается, и, возможно, добавить нечто новое к имеющемуся дизайну.

В качестве примера используйте `$ boxes -l`, чтобы перечислить имеющиеся примеры дизайна, и

```
$ boxes -d <design_name> input.txt output.txt
```

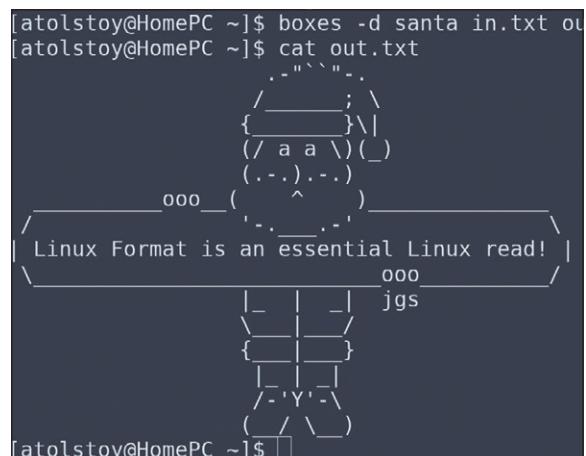
чтобы применить желаемый стиль блока.

Реальный пример может выглядеть так:

```
$ boxes -d xes input.txt output.txt
```

Файл с исходным текстом подготовьте заранее, поскольку `Boxes` не принимает вводимый текст напрямую в качестве параметра командной строки. Попробуйте сначала поиграть с небольшими фрагментами текста — с дизайном `Boxes` они выглядят

## Этот инструмент размещает вокруг текста блоки псевдографики.



➤ `Boxes` позволяют вам добавить в ваш текст украшения, например, Деда Мороза — его-то в апдреле и не хватало!

лучше всего. Большие фрагменты текста заставляют блок слишком сильно растягиваться, что убивает элегантность дизайна. К счастью, есть множество полезных опций, способных улучшить результат. Например, есть опция `-s` для настройки разметки блока вручную, примерно так:

```
boxes -d xes -s 1200x60 input.txt output.txt
```

Предусмотрены также специальные опции для заполнения и отступа, которые можно использовать, чтобы блок симпатично разместился вокруг вашего текста. Воспользуйтесь `$ boxes -h output`, чтобы изучить синтаксис.

## HotGames Развлекательные приложения

### Аркада

# Supertux

Версия: 0.4 Сайт: <http://supertuxproject.org>

**П**ервым делом многие обнаруживают в *Supertux*, что это вовсе не *Supertuxkart*. Эти две игры абсолютно разные: *Supertux* — клон *Super Mario* с графикой, персонажами и музыкой, измененными ради талисмана Linux и зимней темы. Игре уже почти 13 лет, однако в 2005 г. ее разработка почти прекратилась и с тех пор движется очень медленно, но она всё равно имеется почти во всех дистрибутивах Linux. После десяти лет бездействия у нас появился новый стабильный релиз — *Supertux 0.4*. Хотя к тому времени, когда вы будете это читать, вы сможете установить эту версию из ваших репозиториев по умолчанию, есть также сервер со статическими сборками, независимыми от дистрибутива, на <https://lutris.net/files/games/supertux>.

Сама игра была довольно сильно переделана, и в ней переписан графический

стек, под использование OpenGL. Имеется также встроенный в игру менеджер для скачиваемых дополнений и переводов, много новых уровней и карт, новые визуальные эффекты (блеск, частицы и т.д.) и сотни мелких улучшений.

В игре Тукс [Tux] прокладывает дорогу по ледяной Антарктике, заодно сметая плохих парней и направляясь к замку Нолок [Nolok], чтобы спасти Пенни [Penpy], свою возлюбленную. По пути Туксу нужно истребить или избежать разных врагов, которые различаются в зависимости от «мира», в котором вы играете (мир, как и в *Mario*, является набором разных



Если вы устали от бесконечных ледяшек и снежков, попробуйте мир Леса в *Supertux*.

уровней). Новый *Supertux* не придерживается набора персонажей или миров, связанных со снегом; в главном меню игры присутствуют такие миры, как Лес [Forest] или Тропические острова [Tropic islands].

Игра использует современную ускоренную графику через SDL 2 и поддерживает множество контроллеров игры — что необходимо во избежание набивания вам мозолей на больших пальцах от постоянного нажатия на пробел для прыжков.

### Графический стек переписан под ис- пользование OpenGL.

### Головоломка-паззл

# Biniax

Версия: 1.2 Сайт: [www.biniax.com](http://www.biniax.com)

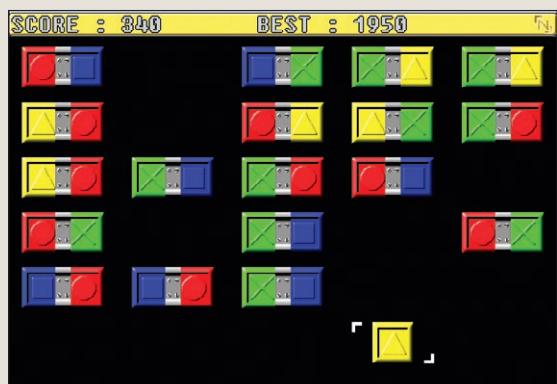
**М**ножество пазлов выглядят винтажно, но лишь некоторые из них предлагают уникальную механику игры. *Biniax* — редкая птица, и предлагает весьма необычный геймплей. Игра появилась в 2005 г. и быстро завоевала популярность в сообществах и Windows, и Linux.

Текущая версия Linux основана на SDL и очень аккуратно портирована с Windows на Linux почти тогда же, когда ее код вышел под лицензией GPL.

Игровой процесс исключительно прост — вы перемещаете свой блок и пробиваетесь вперед. Есть несколько типов блоков: зеленый крест, голубой квадрат, желтый треугольник и красный круг. Блок, которым вы управляете, случайным образом превращается в другой тип после каждого столкновения, и целью игры является ударить по паре блоков своим блоком, и каждый удар уничтожает

пару. Вы можете ударить только пару, содержащую блок, который в данный момент имеется у вас, и каждые три секунды все блоки смещаются вниз, к низу экрана. Как и в Тетрис, игра заканчивается, когда больше не остается места для движения, но *Biniax* не так прост, как может показаться. Если игрок спешит очистить экран от пар блоков, легко попасть в ситуацию, когда вы не можете достичь ни одной пары, содержащей нужный вам тип блока, поскольку вы заблокированы другими парами. В этом случае игрок все равно может перемещать свой блок, пока для него не появится место, но в этот момент игра уже практически проиграна.

### Отличное упражне- ние для развития реакции и смекалки.



Biniax: навечно пропал с этим желтым треугольником...

*Biniax* — отличное упражнение для развития быстрой реакции и смекалки, где вам надо не просто спешно двигать блоки, но и оценивать обстановку на экране, принимая верные решения, чтобы избежать ловушек. Опций в меню игры, сверх обычных New [Новая], Continue [Продолжить] и Quit [Выход], немного, но она явно привлечет внимание любителей головоломок.

## Браузер документации

# Zeal

**Версия:** 0.1.1 **Сайт:** <https://github.com/zealdocs/zeal>

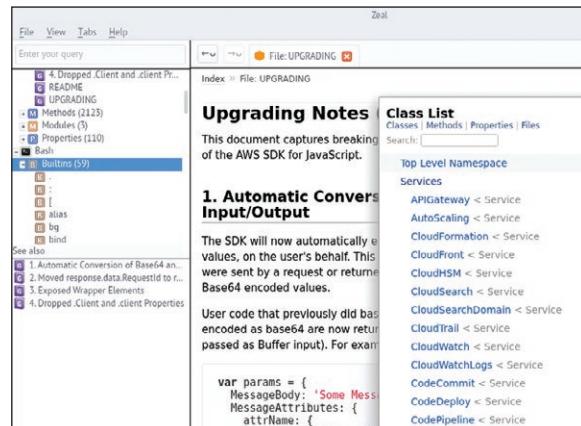
**В**сем бывает надо искать информацию в Интернет или просматривать официальную документацию при работе с языком или средой программирования. Те, кто начинает изучать программирование и знакомится с разработкой программ, выполняют *большой* объем поиска, чтобы изучить основы, но даже опытные разработчики ощущают сильный дискомфорт, когда соединение становится очень плохим и они не могут проверить то, чего не помнят наизусть.

*Zeal* — довольно простой офлайн-браузер документации, вдохновленный такими проектами, как *Dash* (OS X, iOS), *Velocity* (Windows) и *LovelyDocs* (Android). *Zeal* очень похож на вышеперечисленные программы, и, как и другие надстройки документации, он делит базу данных документации, руководства API и фрагменты кода с *Dash*, который появился первым. Вся база данных свободная, и *Zeal* является приложением Linux с открытым кодом, которое может получить к ней доступ.

*Zeal* внешне напоминает встроенные браузеры подсказки в KDE, Gnome и даже в Windows: левая часть хранит дерево содержимого, а основная часть отображает содержимое текущего раздела.

При первом запуске *Zeal* его окно кажется пустынным и голым. Чтобы приступить к работе, надо перейти в File > Options > Docsets [Файл > Опции > Наборы документов] и скачать набор-другой документов, отметив в списке нужное и нажав на кнопку Download [Скачать]. Когда вам покажется, что пора обновить материалы, которые вы скачали раньше, то на это предусмотрена кнопка Update Docsets [Обновить наборы документов] или Refresh [Обновить] для всего индекса имеющихся наборов документов.

**Zeal выглядит так же, как встроенные браузеры в KDE.**



› **Zeal** очень охотно помогает вам найти ответы.

Наборы документов хранятся в директории `~/local/share/Zeal/Zeal/docsets`, но вы можете и сами задать директорию, во вкладке General [Общие] в опциях *Zeal*. Каждый набор документов — это набор файлов HTML с частями JSON и SQLite 3 для индексации.

При просмотре раздела в *Zeal* вы всегда можете переключиться на свой браузер по умолчанию, нажав на кнопку Open URL [Открыть URL]. Однако мы считаем, что в самом *Zeal* есть лучшее решение в виде раздела See Also [См. также], где есть соответствующие части, хорошо продуманный интерфейс со вкладками и кнопками навигации.

## Менеджер архивов

# PeaZip

**Версия:** 5.9.1 **Сайт:** [www.peazip.org](http://www.peazip.org)

**P**eaZip — это интерфейс для *P7Zip*, порта Linux для популярного менеджера архивов в Windows, *7Zip*. PeaZip возвращается в HotPicks, и мы хорошо помним, как восхищались этим чудесным приложением в LXF149 и LXF175, так что спустя пару лет прогресса пора взглянуть, что же появилось новенько в недавнем релизе PeaZip.

Первое, что вам надо знать о PeaZip — то, что ему безразличны и дистрибутивы, и рабочий стол, так что имеются сборки для *Qt* и *GTK* и на сайте проекта, и во многих популярных дистрибутивах Linux. Более того, PeaZip предлагается в виде переносимой сборки. Вы просто скачиваете ZIP-файлы, распаковываете их и командуете `./peazip` из его директории, и PeaZip запустится без всяких проблем.

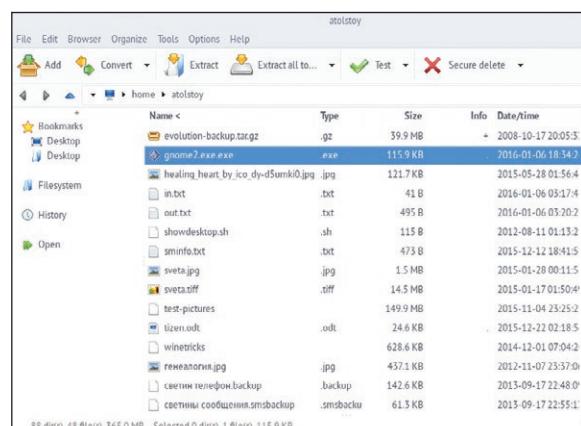
Главное окно напоминает обычный менеджер файлов, и в некотором смысле PeaZip и работает, как менеджер файлов: он позволяет вам запросто просматривать

архивы, как если бы они были обычными директориями.

Еще одна мощная функция — фильтры для просмотра больших архивов. Не секрет, что многие другие утилиты только имитируют «прозрачный» просмотр внутри архива; на заднем плане они должны сначала распаковать архив в некое временное пристанище. PeaZip в этом смысле намного быстрее — вы можете спокойно открыть большой архив с помощью менеджера архивов за несколько секунд.

За недавнее время PeaZip изрядно продвинулась; например, он поддерживает аж 188 форматов архивов, и благодаря использованию самого свежего движка P7zip ему больше не нужна библиотека *UnRAR* для

**Теперь PeaZip поддерживает аж 188 форматов архивов.**



› Понравив примерно минуту с PeaZip, мы уже не можем удержаться от архивирования всех файлов на своем пути...

работы с архивами RAR. Здесь также имеется масса вспомогательных функций, таких, как расширенные программы проверки и анализа файлов, менеджер паролей и т.д. PeaZip также умеет создавать самораспаковывающиеся архивы и даже разделять их, и поскольку это кросс-платформенное приложение, он может создавать архивы в формате EXE в Linux. Мы провели, распаковываются ли они автоматически в *Wine*, и они работают прекрасно. **LXF**

# На диске

Дистрибутивы, приложения, игры, книги и всякое-разное...

Лучшее из Интернета, упакованное в 9 ГБ качественного DVD.



## Дистрибутивы

В этом выпуске Linux Format есть учебник по работе с роутером в Linux. Однако, заканчивая работу над ним, я сделал кое-что, о чем сам же там и предупреждал, и убил роутер, использованный для тестирования. После долгого и упорного изучения форумов и wiki я нашел решение в виде видео на YouTube, где внятно объяснялось, как загрузить новую прошивку, используя режим восстановления роутера и TFTP. Автор взял на себя труд разобрать роутер и присоединить серийный адаптер USB к материнской плате роутера, чтобы точно показать, что происходит, и затем намеренно убил свой роутер, чтобы продемонстрировать, как его исправить. Это была отличная разыскная работа, с отличным объяснением. У автора явно страсть к подобного рода вещам. Конечно, мы привыкли к тому, что в сообществах Linux много увлеченных и готовых помочь людей — однако у этого парня стоит Windows. То есть я не мог использовать программы, о которых он говорил; однако найти им альтернативу не составило труда. Мне пришло в голову, что мы очень гордимся своими сообществами и взаимопомощью, однако вне наших сообществ есть не менее отзывчивые и увличенные люди, которых мы склонны недоценивать исключительно из-за их выбора ОС.

Neil

## » Важно внимание!

### Порченые диски

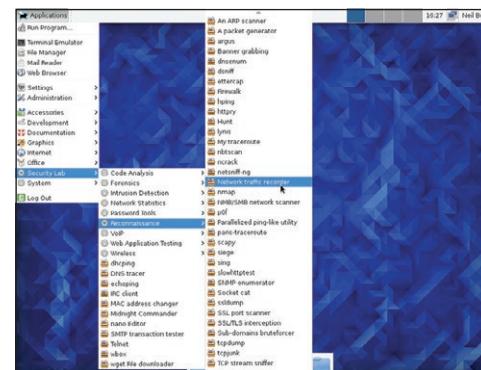
В маловероятном случае какого-то дефекта вашего LXFDVD обращайтесь, пожалуйста, по адресу [disks@linuxformat.ru](mailto:disks@linuxformat.ru) или телефону +7 (812) 309-0686.

Респин безопасности Fedora 23

## Fedora Security Lab

В этом месяце основная тема — тестирование безопасности, и мы подумали, что для вашей работы не помешает соответствующий дистрибутив, а то и два. Fedora Security Lab — это респин Fedora 23, который предоставляет безопасную среду тестирования для работы с ревизией защиты, криминалистической экспертизой и восстановлением системы. Это дистрибутив Fedora 23, где Gnome замещен рабочим столом Xfce, чтобы сделать его легким и быстрым. В нем установлено множество программ безопасности, криминалистического ПО и ПО для тестирования, и все они готовы к использованию.

Вы можете запустить его прямо с DVD или скопировать ISO-образ на флэшку USB, чтобы получить карманную лабораторию безопасности.



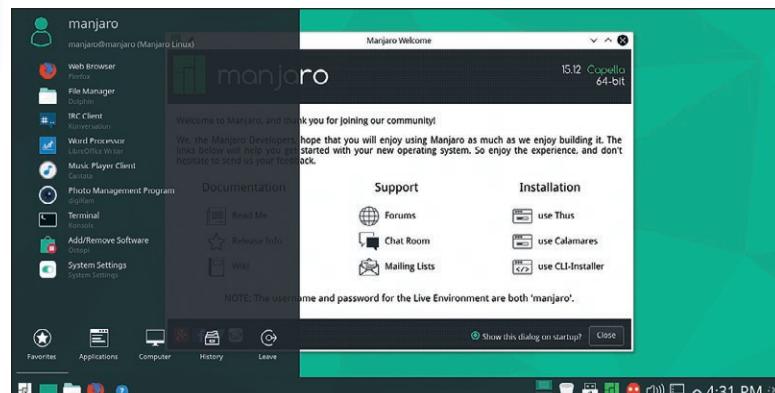
Дружелюбный дистрибутив на базе Arch

## Manjaro KDE 15.12

Одни дистрибутивы, например, Ubuntu, возникают из ниоткуда и практически мгновенно завоевывают популярность. Другие создаются долгое время, постепенно достигая зрелости и развиваясь, пока не наберут популярности и не начнут привлекать больше разработчиков.

Manjaro относится к последним. Он существует с середины 2011 г., и за последние четыре года превратился в достойнейший дистрибутив. Manjaro — один из распущего числа дистрибутивов, основанных на Arch Linux, а не старое производное Debian или Ubuntu. Это

дает ему преимущество доступа к репозиториям пакетов Arch, позволяя обновляться до самых свежих версий программ за весь жизненный цикл дистрибутива, а не ждать появления очередной версии самого дистрибутива. Manjaro дорос до такого состояния, что обзавелся своими репозиториями, и у пользователей есть выбор — работать с уже опробованными и проверенными программами или предпочесть более передовые и новаторские. Доступны разные рабочие столы; мы включили версию KDE, которая, как во многих дистрибутивах, выглядит не похожей на чей-то другой KDE.





# 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 тестирования на вторжения

## Kali Light 2.0

В соответствии с упором данного номера журнала на аспекты безопасности, у нас припасен и второй подобный дистрибутив для вас — Kali Linux, который появился на свет благодаря тем же людям, которые подарили нам BackTrack Linux. Это дистрибутив предназначен для тестирования на несанкционированные проникновения.

В полном релизе имеется рабочий стол Gnome и множество других инструментов, но на DVD для этого оказалось недостаточно места. Это Kali Light с рабочим столом Xfce и самыми важными программами безопасности. Среди них — Metasploit, Nmap, Wireshark, John the Ripper и ряд других. Установив Kali Linux на свой жесткий диск, вы сможете по мере надобности добавить программы.

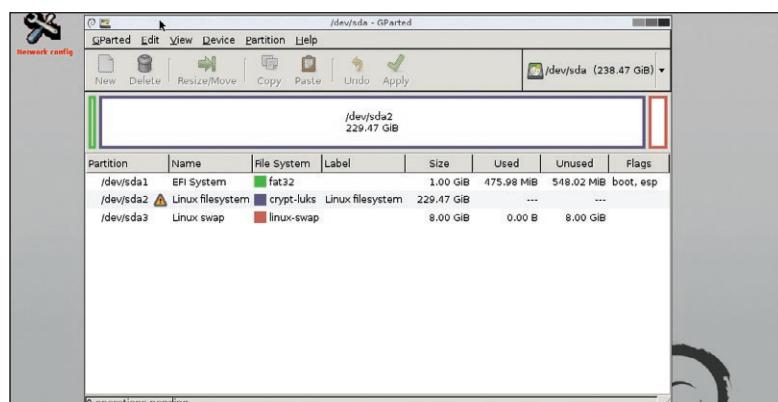


Инструмент разбиения диска

## GParted 0.24.0

Если вы хотите установить на свой жесткий диск более одного дистрибутива Linux, как это объяснялось в нашем материале по множественной загрузке (см. стр. 42), вам нужен способ переразбить свой жесткий диск на разделы. Программы установки дистрибутивов включают инструменты деления диска на разделы, но в основном предназначены для установки на ваш диск своего дистрибутива, и отнюдь не интуитивны. GParted Live — это live CD, созданный на превосходной программе деления диска на разделы GParted. Большинство дистрибутивов включают в свои репозитории

GParted, но если вы хотите изменить системный раздел, не стоит с него загружаться, а live CD — идеальный способ это сделать. С помощью GParted вы можете создавать, удалять, изменять размер и переформатировать разделы по своему желанию, не переживая о влиянии рабочей ОС. Этот 32-битный дистрибутив прекрасно работает на 64-битных системах, и это универсальный инструмент управления диском. Это легковесная система, загружающая минимальный рабочий стол, где уже работает GParted. Загрузите ее с DVD или скопируйте отдельный ISO-образ на CD или флешку USB. [LXF](#)



И еще!

**Системные инструменты**

### Главное

**Checkinstall** Установка tar-архива с помощью менеджера пакетов.

**GNU Core Utils** Основные утилиты, обязательные присутствовать в каждой операционной системе.

**Hardinfo** Инструмент тестирования производительности системы.

**Kernel** Исходный код самого последнего стабильного релиза ядра.

**Memtest86+** Проверьте ОЗУ на предмет неисправностей и сбоев.

**Plop** Простой менеджер загрузки для запуска ОС с CD, DVD и USB.

**RaWrite** Создавайте загрузочные дискеты 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 скриптом Bash** Изучите написание скриптов еще лучше.

**Руководство Bash для начинающих** Освойте написание скриптов Bash.

**Руководство по созданию скриптов Bourne Shell** Начинайте осваивать скрипты оболочки.

**The Cathedral and the Bazaar** [Собор и Базар] Классический текст Эрика С. Реймонда [Eric S. Raymond], объясняющий преимущества открытой разработки.

**Справочник администратора Debian** Базовое руководство для системных администраторов.

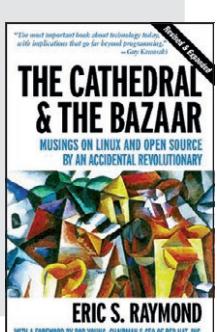
**Введение в Linux** Удобное руководство со множеством подсказок для начинающих пользователей Linux.

**Словарь Linux** Linux от А до Я.

**Linux Kernel in a Nutshell** [Ядро Linux в Двух Словах] Введение в ядро, написанное великим магистром ядра Грегором Кроахартманом [Greg Kroah-Hartman].

**Руководство системного администратора Linux** Полностью контролируйте свою систему.

**Обзор инструментов** Использующий обзор инструментария GNU.



# Пропустили номер?



Закажите его через сайт [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!

Прямо сейчас для заказа доступны следующие номера:

**LXF204**  
Январь 2016

250 руб.

- » Потоки вещания Домашние медиа-центры
- » Быть в теме Синхронизируем свои девайсы
- » Кино и Linux Свободная 3D-графика в Голливуде
- » Печатаем и сканируем Подружим Linux с периферией

**LXFDVD:** Kodibuntu, OpenELEC, openSUSE, OSMC, Ubuntu 15.04, Q408, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: [shop.linuxformat.ru/lxf\\_204/](http://shop.linuxformat.ru/lxf_204/)  
PDF-версия: [shop.linuxformat.ru/elxf\\_204/](http://shop.linuxformat.ru/elxf_204/)

**LXF205/206**  
Февраль 2016

250 руб.

- » Умывальников начальник Управляем умным домом
- » Видеоплееры Лучше семь раз увидеть
- » Raspberry Pi Zero Компьютер за 5 долларов!
- » Организация данных Диски и как с ними бороться

**LXFDVD:** Fedora, Korora, Kubuntu, Tails, Ubuntu, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: [shop.linuxformat.ru/lxf\\_205-206/](http://shop.linuxformat.ru/lxf_205-206/)  
PDF-версия: [shop.linuxformat.ru/elxf\\_205-206/](http://shop.linuxformat.ru/elxf_205-206/)

**LXF207**  
Март 2016

250 руб.

- » Разбегаемся из Окон Прощай, Windows!
- » Резервное копирование Чтобы не было беды
- » Говорят художники Свободным искусствам — свободные инструменты
- » Нас хранит богиня Кали Тесты на вторжение

**LXFDVD:** Linux Mint, openSUSE, Clonezilla, ROSA, Sabayon, Tiny Core, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: [shop.linuxformat.ru/lxf\\_207/](http://shop.linuxformat.ru/lxf_207/)  
PDF-версия: [shop.linuxformat.ru/elxf\\_207/](http://shop.linuxformat.ru/elxf_207/)

Подпишитесь на печатную версию журнала через [www.linuxformat.ru/subscribe](http://www.linuxformat.ru/subscribe) или [www.linuxcenter.ru](http://www.linuxcenter.ru), и получите электронную версию в подарок!

На сайте [shop.linuxformat.ru](http://shop.linuxformat.ru) вы также сможете приобрести предыдущие выпуски LXF.

Телефоны отдела подписки

- » Санкт-Петербург: (812) 309-0686
- » Москва: (499) 271-4954

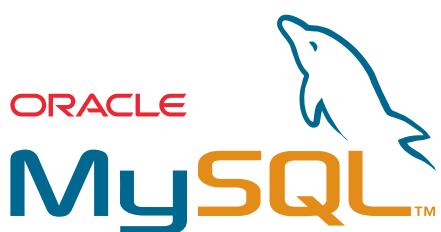


Linux Format ВКонтакте: [vk.com/linuxform](http://vk.com/linuxform)  
Вступайте в нашу группу!

MySQL — одна из самых популярных систем управления базами данных с открытым кодом

Оформите подписку на глобальную техническую поддержку Oracle для продуктов линейки MySQL, и вам будут доступны:

- 7500 специалистов в режиме 24×7
- Неограниченное количество запросов на обслуживание через Интернет или по телефону
- Патчи и обновления на портале MyOracleSupport.com
- Опыт использования продукта более чем у тысячи клиентов



+7 812 309 0686  
[WWW.LINUXCENTER.RU/SHOP/MY\\_SQL](http://WWW.LINUXCENTER.RU/SHOP/MY_SQL) ● [INFO@LINUXCENTER.RU](mailto:INFO@LINUXCENTER.RU)

# Содержание



## Страна 1

### 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.

**Внимательно прочтите это перед тем, как использовать LXFDVD!**

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 bzzip2 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)

#### Дистрибутивы

##### Fedora 23 Security Lan (32-битный)

Расприн Fedora 23, представляющий безопасную среду тестирования для работы с ревизией защиты, криминалистической экспертизы и восстановления системы. Легкий и быстрый благодаря работе чистому ядру Хоса. Запускается прямо с DVD или можно скопировать ISO-образ на USB-брелок.

##### Manjaro KDE 15.12 (64-битный)

Быстрая, удобная настольная ОС на базе Arch Linux. Синтетический процессом установки, автоматическим обновлением оборудования, и стабильным графиком обновлений. Пользователи могут также извлечь выгоду от поддержки активного сообщества форума Manjaro.

##### Kali Linux 2.0 (64-битный)

Дистрибутив на базе Debian с набором инструментов безопасности и криминалистической экспертизы. Представлен в версии без излишеств, недостающие программы можно добавить после установки.

#### НОТРИКС

##### Blinax 1.2 (Головоломка-паззл)

[www.blinax.com](http://www.blinax.com)  
Boxes 1.1.2 Инструмент командной строки

##### Let's Encrypt 0.11 Источник сертификатов

<https://letsencrypt.org>  
Mixxx 2.0 Аудиомиксер и редактор

##### MP3ts 0.91 Конвертер аудио

<http://klientikts-dittiprj.ru/mp3ts>

##### Mumble 1.2.11 Клиент VoIP

[www.mumble.info](http://www.mumble.info)  
Pezzip 5.31 Менеджер архивов

##### Qemu 2.5 Программа виртуализации

[www.qemu.org](http://www.qemu.org)  
<http://www.qemu.org/Download>

Окончание на обороте »

**GParted 0.24.0 (32-битный)**  
Легковесная система, загружающая минимальный рабочий стол, где уже работает Gparted — превосходная программа разделения диска на разделы. Загрузите ее с DVD или скопируйте отдельный ISO-образ на CD или USB-брелок.

# Fedor<sup>a</sup> Security Suite

32-битный

С такой обороны вы будете в безопасности

- » Сканирование сети
- » Тесты на вторжение
- » Взлом сайтов
- » Атаки грубой силы

#### Плюс!

### Kali Light 2.0

Инструмент крутого хакера



# Manjaro 15

**LIVE-ДИСК: ГОТОВ К РАБОТЕ**

Все необходимое для старта в Linux

**НОВЫЙ РЕЛИЗ**

Мощный дистрибутив со скользящими обновлениями, любому по силам

64-битный





## Страница 2

### ДОКУМЕНТАЦИЯ: 10 КНИГ О ЦПУНХ (НА АНГЛИЙСКОМ ЯЗЫКЕ)

### АДВАНСЕД БАШ Скриптинг Гайд

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 Раймонда [Eric S Raymond]

«Собор и базар»

### УЧЕБНИКИ

Sonic Pi

Swift

### The Debian Administrator's Handbook

Руководство администратора написанное разработчиками Debian

### Drive Into Python

Учебник по программированию на Python

### Intro to Linux

Начальное руководство по Linux

### Linux Dictionary

Словарь по языку Linux, объясняющий

### Linux Kernel in a Nutshell

Описание ядра Linux, созданное одним из его выдающихся

разработчиков — [Грег Кроан-Hартман]

### GNU Tools Summary

Руководство по работе

### в Командной строке и обзор основных утилит GNU

### ПОМОЩЬ

### Руководство новичка

Руководства

Ответы

ЧВО (FAQ)

### СРАВНЕНИЕ:

### TRANSCODEДРЫ ВИДЕО

### FF Multi Converter 1.7.2

Браузер документации

### Zeal 0.11

Браузер документации

### Handbrake 0.10.2

Менеджер файлов

### FFMpeg 5.1

Менеджер

### Kernel 5.1

Менеджер

### Memtest86+

Менеджер

### Pop

Менеджер

### SBM

Менеджер

### Wvdial

### ГЛАВНОЕ

### CheckInstall

Менеджер

### Coreutils

Менеджер

### Hardinfo

Менеджер

### Kernel

Менеджер

### Memtest86+

Менеджер

### Pop

Менеджер

### SBM

Менеджер

### Wvdial

### КОММЕНТАРИИ

### Призывают к записи

настройки

### для записи

настройки

### ДИСТРИБУТИВЫ

### Bochs 3.1.1

Аппарастик (64-битный)  
Элегантный и легковесный клон Ubuntu, с рабочим

столом Моктой и легендарным Enlightenment-17, характеризуемый модульностью и высокой настраиваемостью.

### LXLE 14.04.04

(64-битный)  
Простой и легковесный настольный дистрибутив

на базе Ubuntu (LTS) с рабочим столом Xfce.  
Хороший выбор приложений по умолчанию

и полезные модификации.

### Robohub Mate V8.4

(64-битный)  
Удобная и интуитивно понятная OS на базе свежей

стабильной версии Debian, с готовым пакетом поддержки VirtualBox с Windows XP или Windows 7

(платная опция, приобретаемая через Интернет-магазин проекта).

Все дистрибутивы представлены ISO-образом, который можно и записать на отдельный носитель, и загрузить в live-режиме прямо с LXLEDVD.

**Пожалуйста, перед использованием  
снимка дистрибутива, пожалуйста,  
с инструкцией, опубликованной  
в журнале на стр. 109!**

**КОММЕНТАРИИ** Пристыгните и поговорите о своем опыте использования Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажмите на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

## Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу — это обратиться к программе *cdrecord*. Для всех перечисленных ниже действий потребуются права суперпользователя-root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

`cdrecord -scanbus`

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке — например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

`cdrecord dev=0,3,0 -v /path/to/image.iso`  
Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле */etc/default/cdrecord*. Добавьте по одной строке для каждого устройства записи (скорее всего, в вашей системе присутствует только одно такое устройство):

`Plextor=0,3,0 12 16M`

Первое слово в этой строке — метка; затем после адреса SCSI-устройства вы должны указать скорость и 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

`cdrecord -v /path/to/image.iso`

Если вы не принадлежите к любителям командной строки, в таком случае вам придется на помощь утилита *gcombust*. Запустите ее от имени root и выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажмите на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

## Другая ОС?

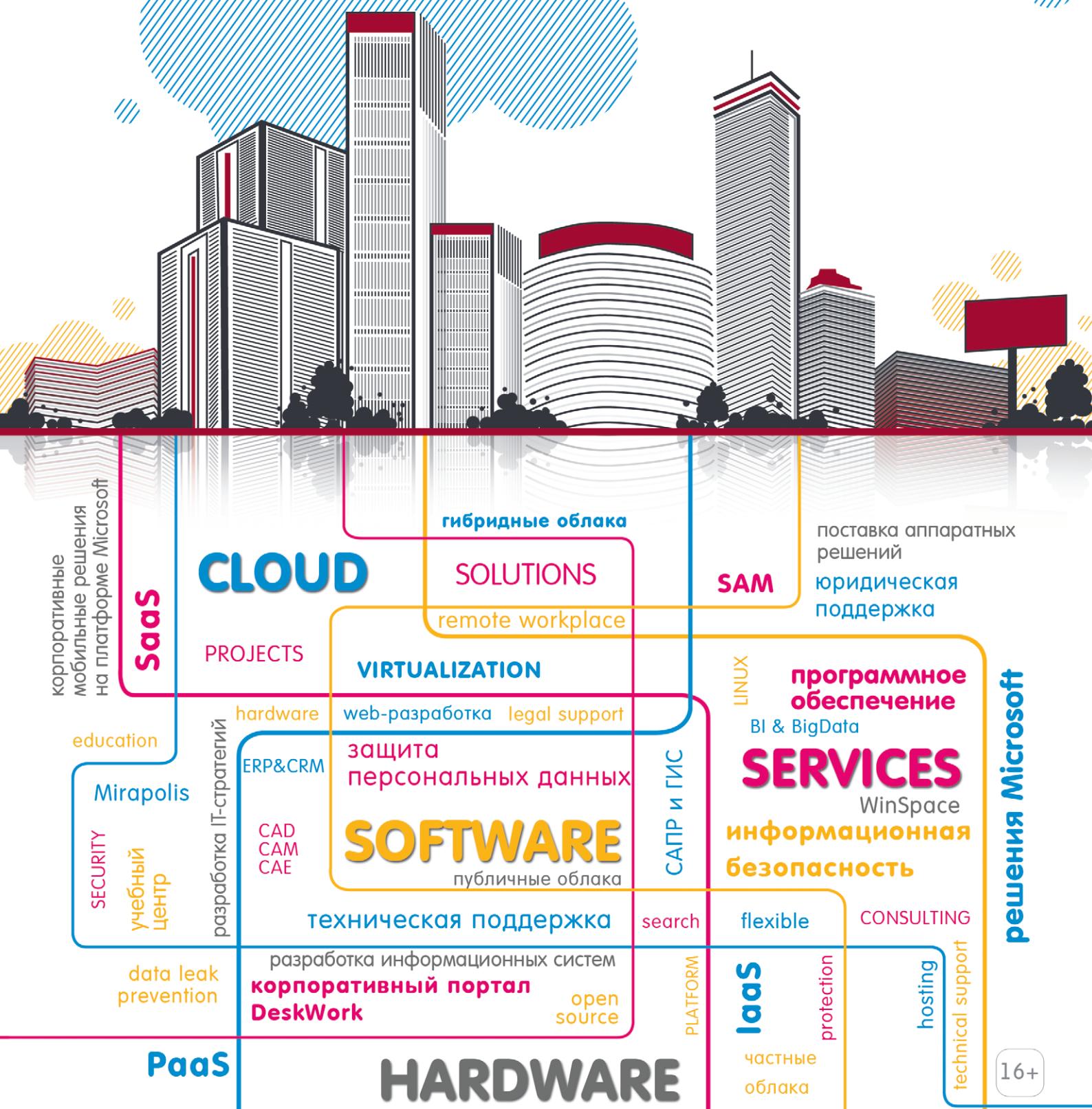
Использовать Linux для записи компакт-диска не обязательно. Программы вроде *cdrecord* просто переносят двоичные данные на чистую матрицу. Все необходимые файлы уже включены в ISO-образ, который распознается любой операционной системой, будь то Linux, Windows, Mac OS X или AmigaOS.

## Нет устройства для записи дисков?

Если у вас нет устройства, с помощью которого можно было бы записать образ на диск, можно найти какого-нибудь друга или организацию, у кого есть компьютер с дисководом, и прокинуть 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.

## IT-архитектура вашего бизнеса





## В майском номере

# Изучите, как... Встраивать!

Linux приведет вас в прекрасный новый мир, встраиваясь во все подряд — от квадрокоптеров до автомобилей.

## BBC micro:bit

Мы наложили лапы на потрясающее новое устройство от BBC и выяснили, как запустить Python и подкрепить его Pi.

## Проекты на Swift

Углубите свое знание Swift и создайте свое первый полноценный web-проект. Мы поможем.

## Планшеты с Linux

Хватайте дешевый планшет и запустите на нем Linux. Описываем все подводные камни и способы добиться гладкой работы.

Содержание будущих выпусков может меняться — вдруг до нас доберутся Вещи из Небезопасного Интернета...

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 2000 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

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/и, лит. Ф

Тел. (812) 462-8383, e-mail: [office@dprint.ru](mailto:office@dprint.ru)

Заказ 14089

### 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 Крис Торнетт [Chris Thornett] [chris.thornett@futurenet.com](mailto:chris.thornett@futurenet.com)

Художественный редактор Эфраин Эрнандес-Мендоса [Efrain Hernandez-Mendoza] [efrain.hernandez-mendoza@futurenet.com](mailto:efrain.hernandez-mendoza@futurenet.com)

### ПОДГОТОВКА МАТЕРИАЛОВ

Джонни Бидвелл [Jonni Bidwell], Нейл Ботвик [Neil Bothwick], Джолион Браун [Jolyon Brown], Джеймс Хант [James Hunt], Аластер Дженнингс [Alastair Jennings], Аллан Мак-Каллах [Allan McCullagh], Чунг Нгуен [Chung Nguyen], Ник Пирс [Nick Peers], Лес Паундер [Les Pounder], Рафик Рехман [Rafeeq Rehman], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Дэнни Торп [Danny Thorp], Александр Толстой [Alexander Tolstoy], Михалис Цукалос [Mihalis Tsoukalos], Бен Уилсон [Ben Wilson], Евгений Балдин, Андрей Гондаренков, Дмитрий Пантелеевичев, Алексей Федорчук, Максим Черепанов, Лада Шерышова

Иллюстрации Шейн Коллиндж [Shane Collinge]

### 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 предоставленными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неизъяснимое право на публикацию и лицензование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всегда, 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 ошибки в материалах.

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 повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торvalдса [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 журналах, издаваемых 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



Виртуальные серверы нового поколения



# Серверы летают!

- ✓ 3 data-центра в России и Европе
- ✓ Высочайшая скорость Enterprise SSD
- ✓ Тройная репликация всех данных
- ✓ Безлимитный трафик
- ✓ Докризисные цены — от 200 рублей!

Закажите прямо сейчас  
[infobox.ru/vps/](http://infobox.ru/vps/)

## Clever Solution.



Все цены указаны без учёта НДС и регулируются условиями предоставления услуг компанией Hetzner Online GmbH. Цены могут быть изменены. Все права защищены соответствующими производителями.

**Выделенный сервер EX51**

Intel® Core™ i7-6700  
Quad-Core Skylake Processor  
64 ГБ DDR4 RAM  
2 x 4 ТБ SATA HDD Enterprise класс  
1 Гбит/с гарантированно  
100 ГБ место для резервных копий  
30 ТБ трафик\*  
Без минимального контракта  
Установка 8200 рублей

**4100**  
рублей в месяц

**Выделенный сервер EX51-SSD**

Intel® Core™ i7-6700  
Quad-Core Skylake Processor  
64 ГБ DDR4 RAM  
2 x 500 ГБ SATA SSD Enterprise класс  
1 Гбит/с гарантированно  
100 ГБ место для резервных копий  
30 ТБ трафик\*  
Без минимального контракта  
Установка 8200 рублей

**4100**  
рублей в месяц

\* Нет платы за превышение. При превышении 30 ТБ/месяц скорость соединения ограничивается (подсчёт ведётся по исходящему трафику, входящий и внутренний трафик не учитывается). Опционально можно снять ограничение, подтвердив оплату 98 рублей за каждый дополнительный ТБ.